Data Recasting for Natural Language Inference on Tables

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computational Linguistics** by Research*

by

Aashna Jena
20171095
aashna.jena@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May, 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Data Recasting for Natural Language Inference on Tables"** by Aashna Jena, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Manish Shrivastava

To Mumma, Papa and Anu.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr Manish Shrivastava, who has guided me throughout my years as a Master's student. I would also like to thank the defense committee and my reviewers, who have taken out the time to go through my work and validated it. A special thanks to Prof Radhika Mamidi and Prof Dipti Mishra for being amazing professors and sparking my interest in Linguistics.

I could not have undertaken this journey without my co-author Vivek Gupta, who has been a hands-on mentor to me and helped me every step of the way in my Master's project. Many thanks to Julian Eisenschlos as well, for a great collaboration experience and much appreciated guidance in times of need. A big shout out to Mihai, my manager, for his unconditional support and for making the last leg of my Master's journey a smooth one.

A big personal thanks to my parents, my sister Anu, and my friends - Debojit, Devesh, Priyank, Abhinav, Suryansh, Shantanu, Souvik, Harshita and Anushka, who have been constant sources of support in my journey. A special mention to Manu and Mrinal, who have been mentors, friends, and care-givers all-in-one for a significant part of my journey. I'd also like to recognise Chaitanya Agrawal, Vamshi Krishna, Shirley, Sreeharsha T D, Pranshu Pandya, Advaith Malladi, Druhan Shah, Eshika Khandelwal and Bhavik Chandna for their assistance in my research work.

Abstract

Given a premise, the aim of Natural Language Inference is to identify a hypothesis as Entailed, Refuted, or Neutral. To do such classification, a model must acquire the ability to reason over the premise. While entailment tasks have been extensively studied with unstructured text as the premise, there is an increasing demand for learning to reason over semi-structured and organised data formats such as tables, knowledge graphs, databases, and combinations thereof. Structured data forms differ from unstructured text in the way they capture information and relationships – not just via language, but also through position and structure. Particularly, tables capture the connections between cells, which represent isolated distinct entities. Tabular data is organised so that items of the same kind are grouped together in rows, columns, or both. Consequently, it is straightforward to infer rankings, trends, unique items, and aggregate values from tabular data. These sorts of reasoning are specific to structured data formats, which makes inference on tables a difficult task requiring separate effort from inference on plain text.

Creating challenging tabular inference data for supervision is necessary for mastering complex reasoning. Prior research in this sector has predominantly employed two data generating methodologies. The first technique is human annotation. This results in data that is inventive, fluent, and linguistically diverse. However, human annotation is costly and time-consuming, making it difficult to scale. The second form of data production is through synthetic means, where the data is generated using a defined set of rules or context-free grammar. This system is easily scalable in terms of both time and cost, but it lacks originality. Its results are predictable and adhere to predetermined patterns and fixed vocabulary.

This research presents a framework for semi-automatically "recasting" existing tabular data in order to mitigate the drawbacks of both of the aforementioned data generation techniques. Existing data is perturbed, modified, and augmented through recasting to conform to the specifications of a given target task, which is Tabular Inference in this case. This framework is used to construct tabular NLI instances from five datasets that were originally designed for tasks such as table-to-text generation, tabular question answering, and tabular semantic parsing.

To demonstrate the utility and quality of these datasets, this thesis explains how recasted data may be utilised as evaluation benchmarks and augmentation data to improve performance on tabular NLI tasks such as TabFact. In addition, this work evaluates the efficacy of models trained on recasted data in the zero-shot setting and examines performance trends across different types of recasted datasets. This thesis concludes with a discussion of the limitations and potential future paths of this field of study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most extensively researched problems in Natural Language Processing is that of Natural Language Inference (NLI). Given a premise, NLI is the task of classifying a hypothesis as Entailed, Refuted or Neutral. In the NLI setting, a premise could be anything from a sentence, to a paragraph, to a document or a collection of documents. Premises could also contain non-textual information, such as images, tables, forms and diagrams. A hypothesis is typically a statement. The task of NLI is to determine whether the hypothesis can be reasonably inferred from the premise. If the hypothesis can be logically deduced solely from the information available in the premise, without any external knowledge source, it is labeled an **Entailment**. Similarly, a false statement is labeled a **Contradiction**. A statement that cannot be deemed True or False solely on the basis of the hypothesis is labeled **Neutral**. Several large scale datasets such as SNLI [4], MultiNLI [80], and SQuAD [61] explore NLI with unstructured text as the premise. Table 1.1 shows an example of NLI on unstructured text. To perform NLI, a model needs to have a thorough compositional understanding of a sentence.

In this thesis, I focus on the task of NLI on tables. Tables are a form of semi-structured data that come with their own set of challenges when it comes to inference. I aim to push the results on an existing tabular NLI task i.e. TabFact [8] through the addition of high-quality large-scale pre-training data. Although TabFact is the downstream task I wish to solve, this thesis focuses on the generation and validation process of the augmentation data. The improved accuracy results on TabFact further verify the utility and quality of the generated data. This chapter introduces the target task, and elaborates on the motivation to solve it through a "Recasting" setting.

#	Premise	Hypothesis	Label
1	A man inspects the uniform of a figure in some East Asian country.	The man is sleeping.	**Contradiction**
2	An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	**Neutral**
3	A soccer game with multiple males playing.	Some men are playing a sport.	**Entailment**

Table 1.1: Example of Natural Language Inference (taken from https://microsoft.github.io/nlp-recipes/examples/entailment/)

1.1 Inference on Semi-structured Data

While inference on unstructured text is commonly researched, structured data forms (e.g. tables, knowledge-graphs and databases) pose a different set of problems. Structured data has the ability to capture relationships between entities through means other than language. For example, tables capture links between cells through their relative positions. In Table 1.2, two cells in a row give information about the same record event, and two cells in a column give the same metric for different records.

For NLI on unstructured text, a model must understand the concepts of presupposition, meronymy, holonymy, hypernymy and hyponymy. For example, in example #3 of Table 1.1, a model must understand that "soccer" is a "game" ("soccer" is a hyponym of "game"). Similarly, the model must also understand that in example #1, the action of "inspecting" presupposes the man to be awake.

In contrast to unstructured text, tabular data opens the doors to reasoning of much more complex types. Tabular data differs from unstructured text in the way that it can capture information and relationships in a succinct manner through underlying structure [26]. On tables, we can reason about ranking, counting, and aggregation. Table 1.2 gives examples of different kinds of entailments derived from the premise. Since understanding tables requires understanding positional structure, traditional language models do not suffice for this task. Approaches along the line of converting tables to paragraphs using templates and applying language models on them have shown limited success. Traditional NLI models also have limited capability to solve complex reasoning types such as counting and aggregation.

Creating challenging large scale supervision data is hence vital for research in tabular reasoning. In recent years, several NLI tasks have been introduced, which use tables as the premise e.g. tabular inference (TNLI) datasets such as TabFact [8], InfoTabS [26] and shared tasks like SemEval 2021 Task 9 [77] and FEVEROUS [3].

Men's Athletics World Records

Event	Perf	Speed(mph)	Athlete	Nat	Date
100 m	9.58	23.35	Usain Bolt	JAM	16 Aug 2009
200 m	19.19	23.31	Usain Bolt	JAM	20 Aug 2009
400 m	43.03	20.79	Wayde van Niekerk	RSA	14 Aug 2016
800 m	1:40.91	17.734	David Rudisha	KEN	9 Aug 2012
1000 m	2:11.96	16.952	Noah Ngeny	KEN	5 Sep 1999
1500 m	3:26.00	16.288	Hicham El Guerrouj	MAR	14 Jul 1998

Hypothesis	Reasoning
Wayde van Niekerk's 400m record is the most recently set record	Superlative
Usain Bolt's speed was faster in the 200m event than in the 100m event	Comparison
4 out of a total of 6 records mentioned in the table were set after 2000	Counting
The record setting athletes hold 4 distinct nationalities	Uniqueness
The average speed of all athletes combined is 19.7 mph	Aggregation

Table 1.2: Table taken from https://en.wikipedia.org/wiki/List_of_world_records_in_athletics

In this thesis, I work on the TabFact [8] dataset. It is a large-scale dataset based on open-domain Wikipedia tables. The human-annotated hypotheses range from simple to complex, depending upon the kind of reasoning they cover. 3 expands on these distributions.

1.2 Motivation

Broadly, there are two distinct approaches to the generation of supervision data. One way is through the use of human annotation, and another is through the use of templates or context-free grammar. I examine the pros and cons of both approaches, and then I suggest a middle ground that makes use of the benefits of both approaches while mitigating the drawbacks to a significant degree. I present the reasons why I believe that this path is the best one, including the several advantages it offers, particularly with regard to tables.

Context free grammar rules

<statement>	→	<expr> <compare> <expr>
<expr>	→	<select> when <where> | <select>
<select>	→	<column> | the <aggr> of <column> | the count
<where>	→	<column> <compare> <value> | <where> and <where>
<aggr>	→	first | last | lowest | greatest | sum | average | range
<compare>	→	is | is greater than | is less than
<value>	→	<string> | <number>

Operation	Reference result
first	the value in given column with the lowest row index.
last	the value in given column with the highest row index.
greatest	the value in given column with the highest numeric value.
lowest	the value in given column with the lowest numeric value.
sum	The sum of all the numeric values in given column.
average	The average of all the numeric values in given column.
range	The difference between greatest and lowest in given column.

Table 1.3: Context free grammar for hypothesis generation from tables. Tables are expected to be vertically oriented with the top row containing headers and each column containing entities of the same type. Taken from Eisenschlos *et al.* [19]

1.2.1 Data Generation Methods

Human annotation has been long used for creating datasets of superior quality. Annotators are given guidelines for creating data for a specific task at hand. For Tabular NLI, the task would be to produce a hypothesis, given a table. Human written hypotheses would be fluent and creative, since every annotator would think differently, and phrase their annotations distinctly. The hypotheses are likely to be coherent sentences with adequate linguistic diversity - both structural and lexical. Linguistic diversity ensures that the model does not overfit to learn only certain keywords or grammatical structures.

Despite the advantages, the costly and time-taking nature of human annotation makes it a tedious task and asks for alternate data generation methods to be explored. Human written data is hard to scale, which is why it cannot always be used to create large scale datasets. Furthermore, Gururangan *et al.* [29] and Geva *et al.* [22] show that many human-annotated datasets for NLI contain annotation biases or artifacts. This allows NLI models to learn spurious patterns [48], which enables models to predict the right label

Example Table

Party	Votes(thou)	Seats
Party A	650	120
Party B	570	89
Party C	TBA	89
Party D	575	95
Total	1235	298

Example Procedure

Source data point (Q/A):

Question: *How many seats did Party B win?*

Answer: *89*

recast

Entailment: *Party B won 89 seats.*

Contradiction: *Party B won 120 seats.*

Table 1.4: Example of Tabular Data Recasting from a Question Answering source dataset

for the wrong reasons, sometimes even with noisy, incorrect or incomplete input [54]. Recently, Gupta *et al.* [27] revealed that tabular inference datasets also suffer from comparable challenges. Furthermore, Geva *et al.* [22], Parmar *et al.* [50] show that annotators introduce their own bias during annotation. For example, Gupta *et al.* [27] demonstrates that annotators only generate hypothesis sentences from keys having numerical values, implying that some keys are either over or underutilized. For reasons mentioned above, we require millions of data points to understand tabular data.

On the other hand, we have automatic generation methods to create data. These use templates or context free grammars to create sentences. Data created through these methods is extremely easy to scale, and is both time and cost effective. One can control the distribution of the data as well. For example, one can control the ratio of labels - **Entailment**, **Neutral** and **Contradiction**. One can also control the ratio of hypotheses produced with different types of reasoning. Data produced is also free of human bias such as over-use or under-use of certain keys in the table. An example of producing table NLI data through context free grammar is shown in Table 1.3.

Data created through such methods is often referred to as "Synthetic Data". This is due to the lack of human involvement in it. Synthetic data, despite its scalability, is not as diverse and creative as human annotated data. Its vocabulary and compositional power is limited to those mentioned in the templates or context free grammar. In a way, we trade off quality for quantity when we move from human annotated data to synthetic data. In my project, I aim to answer the following question : *Can we generate challenging supervision data that is as scalable as synthetic data and yet contains human-like fluency and linguistic diversity?*

1.2.2 Data Recasting

In this work, I 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. In my research, I choose human-annotated source datasets meant for non-NLI tasks and transform them to NLI

data i.e. hypotheses and their appropriate labels. Data recasting is a "middle route" between human annotation and synthetic generation that exploits the advantages of both, while canceling out their disadvantages. Recasted data, or data created through recasting maintains linguistic diversity owing to the human involvement in the creation of source datasets. Since the transformation process is largely automated, recasting strategies are easy to scale. 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. The resultant data checks both requirements - quality and quantity.

Although data recasting has been around for a long time, for example, QA2D [14] and SciTail [36] effectively recast question answering data for inference (NLI), no earlier study has applied it to semi-structured data.

Therefore, I propose a semi-automatic framework for tabular data recasting. Using this framework, I 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). Table 1.4 shows an example of tabular data recasting.

1.3 Viability of Recasting for Table NLI

In this work, I use recasting to create augmentation data for a specific downstream task i.e. TabFact. To be able to show its effectiveness as augmentation data, the recasting process needs to generate data which is similar (in domain, writing style and composition) to the target dataset.

How do we know that a "Recasting Framework" will be able to meet these requirements? While it is almost impossible to find readily available data that matches your target task description, domain and writing style, it is much more probable to find datasets that are sourced from the same domains but perform different tasks. This is the phenomenon that a recasting framework exploits - it uses such similarly sourced datasets to generate data for particular target tasks. In the case of tabular NLI, the semi-structured nature of the premise facilitates the transformation process even more. In our particular case, the target task i.e. TabFact uses Wikipedia tables as the premise, and I find that several tabular-based datasets also source tables from Wikipedia. The TabFact datatset also lists the kind of reasoning it uses for its NLI hypotheses, and I identify that tasks such as Question Answering and Semantic Parsing build over similar kinds of reasoning as well. I find datasets that fall into this intersection of requirements, and apply the recasting framework on them to generate NLI data (chapter 4 covers these datasets in detail).

Recasting has been previously investigated for unstructured text. As established in the beginning of this chapter, structured data representations such as tables vary from plain text in various ways. Let's explore a basic example for understanding how new data can be created for Table NLI. Suppose I have a table and a hypothesis statement, and I wish to alter this statement to generate other assertions. Every statement consists of elements that are directly taken from the table and portions that connect these bits of information using grammatical rules. Table 1.5 illustrates instances of hypotheses and the extent to which their components are obtained from the table. To edit any of these claims, it is necessary to

Example Table

Party	Votes(thou)	Seats
Party A	650	120
Party B	570	89
Party C	TBA	89
Party D	575	95
Total	1235	298

Example Hypotheses

#1 Party B won 89 out of a total of 298 available seats .

#2 Party A managed to secure more votes than Party D .

Entities coming directly from the table, directly affecting the truth value of the statement.

Entities stemming from the table, used for sentence composition.

Unhighlighted parts of the statement are purely grammatical.

Table 1.5: Example of entities derived directly, indirectly and independent of the table

understand which entities impact their truth value. These are the entities which originate straight from the table. For instance, altering 89 to 99 in hypothesis #1 would render the assertion incorrect. This demonstrates that in order to change or disrupt hypotheses, it is necessary to be able to draw alignments between table cells and tokens in the provided hypothesis statement. Here, the arranged form of tabular data facilitates the Recasting procedure.

1.3.1 Advantages of semi-structured data

Tables are arranged in rows and columns, which capture relationships between cells. Moreover, 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. But since we know exactly what constitutes an entity based on the table cell boundaries, we only have to do a 1-way lookup, i.e. match cell contents with n-grams in the statements.

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. This is incredibly beneficial when modifying source data by substituting entities. For example, if we aim to create a contradiction out of Hypothesis #1 by changing Party B , we would know exactly which column to look at for finding contradicting values (Party A, Party D etc).

Furthermore, since data is organized, it is possible to find the data type for each column. Frequently, the column header also indicates the data type (e.g. Name, Organization, Year etc). When forming new statements from templates, data type information helps in identifying which operations can be applied to which entities. For example, aggregations are possible only for numeric data, while counting can be done for all kinds of data - one can count the number of rows containing a particular string, number, date or alphanumeric sequence.

Task	Source Datasets
Table to Text Generation	ToTTo [49]
Table Question Answering	WikiTableQuestions [51], FeTaQA [44]
Semantic Parsing on Tables	Squall [68], WikiSQL [92]

Table 1.6: Source datasets used for creating tabular NLI data

1.4 Contributions

In this work, I propose a semi-automatic framework for tabular data recasting. Using this framework, I 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.

The recasted data can be used for both evaluation and augmentation purposes for tabular inference tasks. I choose TabFact [8] as the downstream task to report accuracies. TabFact [8] is a benchmark Tabular inference dataset with binary labels - Entail and Refute. Models pre-trained on the recasted data show an improvement of 17% from the TabFact baseline [8] and 1.1% from Eisenschlos *et al.* [19], a synthetic data augmentation baseline. Additionally, I train models only on recasted data, without fine-tuning on TabFact. Since these models are NLI models in themselves, I report their zero-shot accuracies. I observe a best accuracy of 71.1% on TabFact validation set, which is 5% percent higher than the supervised baseline accuracy reported by Chen *et al.* [8]. The main contributions in this work are the following:

1. I propose a semi-automatic framework to generate tabular NLI data from other non-NLI tasks such as Table2Text generation (T2TG), Tabular Question Answering (TQA), and Semantic Parsing on Tables (SPT).
2. I build five large-scale, diversified, human-alike, and complex tabular NLI datasets sourced from datasets as shown in Table 1.6.
3. I present the usage of recasted data as TNLI model evaluation benchmarks. I present a detailed analysis of how existing models perform on different reasoning categories and on counterfactual data.
4. I present zero shot accuracies of NLI models trained on recasted data and tested on the TabFact test set. I compare and analyse performance of models trained on data derived from different source tasks. I also anaylse the correlation between dataset size and downstream performance.

5. I demonstrate the use of recasted data for augmentation, and show improvements in accuracies for fine-tuned models on the TabFact task. I report the accuracies for simple and complex splits of the test set, and analyse the trends.

1.5 Organization of Thesis

This thesis is organized into 5 chapters. After this introductory chapter, Chapter 2 talks about prior work done in the field of Natural Language Inference and in topics like pre-training, multi-task learning and inference on semi-structured data. After discussing the background of the problem, I explain the crux of my work, the Recasting Framework, in Chapter 3. This chapter derives the prerequisites required for data recasting in the tabular data setting, and outlines a generic step-by-step method to recast data. Chapter 4 applies this data recasting strategy on 5 datasets, as listed in Table 1.6. The sections explains the unique challenges of each dataset in detail, and explains why and how each dataset fits into the recasting pipeline. I also reason for my choice of source datasets. Chapter 3 and 4 detail the experimental setting, report results and perform a deep analysis on the datasets. I perform experiments to answer 3 distinct research questions. Chapter 4 provides an ablation study for the models trained on recasted datasets. Chapter 5 concludes my work and talks about future directions which are outside the scope of this thesis. I discuss the major takeaways from my work.

Chapter 2

Related Work

2.1 Natural Language Inference

Natural language inference, also known as textual entailment, is a text understanding task that has been investigated extensively and features multiple datasets of varying sizes. The practise of recognising entailment dates back a long way in NLP [12]. Several thousands of human-annotated entailment pairs were associated with the annual PASCAL RTE challenges (Dagan *et al.* 11, among other references). The SNLI dataset was created by Bowman *et al.* 4 and is the first large-scale entailment dataset to employ image captions as premises. On the other hand, the MultiNLI dataset was created by Williams *et al.* 80 and incorporates premises from a variety of other domains. The SQuAD question answering data [61] and the Winograd Schema Challenge data [39] were converted into inference tasks in order to create the QNLI dataset and the WNLI dataset, respectively. These datasets offer a fresh perspective as a result, similar to the recasting I do as part of this research. More recently, SciTail [36] and Adversarial NLI [47] have focused on building adversarial datasets; the former uses information retrieval to select adversarial premises, while the latter uses iterative annotation cycles to confuse models. Both of these approaches use information retrieval to select adversarial premises. This is similar to the "counterfactual" data I discuss in this research.

In recent times, difficult new datasets that place an emphasis on complex reasoning have been released. The challenge presented by Qin *et al.* 58 is to identify the most reasonable inferences that may be drawn from observations (abductive reasoning). A significant amount of work concerning various types of reasonings has been published over the entirety of NLP. Common sense reasoning [70], temporal reasoning [94], numerical reasoning (Ravichander *et al.* 63; Wallace *et al.* 75), and multi-hop reasoning [35] are just a few of the types of reasoning that have attracted a lot of attention from researchers.

2.2 Inference on Semi-Structured Data.

Recent developments have allowed the text to text framework to accommodate structured data in the form of knowledge graphs [74], tables [26]), and images [69]. One such illustration is provided by

the comprehensive TABFACT dataset [8]. Typically BERT-based models, which operate on flattened versions of table and employ textual templates to make the tables look like natural language perform exceptionally well in this task.

InfoTabs [26], another tabular inference dataset, explores inference on Wikipedia InfoBoxes. These are tables containing <key,value> type artifacts, which are different from database-like tables containing multiple rows and multiple columns. WikiTableQuestions [52], WikiQAA [1], FinQA [10] and HybridQA [6] perform question answering on tables. Some involve short form question-answering, which requires identifying the answer cells from tables, while some are domain specific or involve long form question answering, which require a model to not just identify relevant cells but also generate answer statements. ToTTo [49], Yoran *et al.* [88], LogicNLG [7] and Logic2Text [9] 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 [24, 72, 15]. Recent work like TAPAS [30], TAPAS-Row-Col-Rank [18], TaBERT [86], TABBIE [31], Tables with SAT [90], TabGCN [56] and RCI [25] use more sophisticated methods of encoding tabular data. TAPAS [30] encodes row/column index and order via specialized embeddings and pre-trains a MASK-LM model on co-occurring Wikipedia text and tables. Yang & Zhu [84] decomposes NLI statements into subproblems to enhance inference on TabFact.

As discussed before, inference on tables includes numeric and logical reasoning. Numeric reasoning in Natural Language processing has been recognized as an important part in entailment models [65] and reading comprehension [62]. In tables, since data is structured and ranked, numeric reasoning becomes a natural part of tabular entailment. Wallace *et al.* 75 studied the capacity of different models on understanding numerical operations and showed that BERT-based models still have headroom. This motivates the use of the data augmentation approaches to improve numerical reasoning in our model. My research builds on this claim, and leads to similar conclusions.

2.3 Data Augmentation

Generating cheap and scalable data for the purpose of training and evaluation has given rise to the use of augmentation techniques. Synthetic data generation for augmentation for unstructured text is explored in Alberti *et al.* [2], Lewis *et al.* [40], Wu *et al.* [81], Leonandya *et al.* [38], and for Tabular NLI is shown in Geva *et al.* [23], Eisenschlos *et al.* [19]. Salvatore *et al.* [64] and Dong & Smith [18] generate synthetic data for evaluation purposes.

Closer to our work, Sellam *et al.* [66] use perturbations of Wikipedia sentences for intermediate pre-training for BLEURT(a metric for text generation) and Xiong *et al.* [82] replace entities in Wikipedia by others with the same type for a MASK-LM model objective. This is similar to the Counterfactual data generation I discuss in this research. I take advantage of other rows in the table to produce plausible negatives, and also replace dates and numbers. Kaushik *et al.* [33], Gardner *et al.* [20] 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.* [43] demonstrate that adding counterfactual hypotheses enhances model performance on the TabFact dataset.

It has been demonstrated that synthetic data can enhance learning in NLP tasks [2, 40, 81, 38], and Tabular NLI in specific [23, 19]. Additionally, Salvatore *et al.* [64] and Dong & Smith [18] generate synthetic data for evaluation purposes. Salvatore *et al.* 64 employ synthetic data generated from logical forms to evaluate the performance of textual entailment models (e.g., BERT). Geiger *et al.* 21 employ synthetic data to construct fair assessment sets for natural language inference. Geva *et al.* 23 demonstrate the significance of incorporating numerical reasoning via generated data into the model in order to solve reading comprehension challenges. They propose different templates for generating synthetic numerical examples. Eisenschlos *et al.* 19 employs a strategy that is better suited for tables and to the entailment task, and is arguably simpler, using a context free grammar.

2.4 Data Recasting

Data generation through recasting has been previously explored for NLI on unstructured data. White *et al.* [79] use semantic classification data as their source. Multee [73] and SciTail [36] recast Question Answering data for entailment tasks. Demszky *et al.* [14] proposes a framework to recast QA data for NLI for unstructured text. Poliak *et al.* [55] presents a collection of recasted datasets originating from seven distinct tasks. For tabular text, Dong & Smith [18] present an effort to re-use text generation data for evaluation.

2.5 Parsing Tabular Data

Using an encoder-decoder strategy, most semantic parsing models are taught to create gold logical forms (Jia & Liang 32; Dong & Lapata 17). Typically, models are trained with weak supervision in the form of denotations to reduce the cost of gathering full logical forms. These are utilised to direct the search for appropriate logical forms.

Other publications have proposed end-to-end differentiable models that are trained under inadequate supervision, but do not generate logical forms directly. Neelakantan *et al.* 45 offered a sophisticated model that successively predicts symbolic operations across explicitly predefined table segments, whereas Yin *et al.* 87 proposed a similar model where the symbolic operations themselves are learned during training. Their model cannot predict aggregations over table cells.

Finally, pre-training approaches with varied training aims have been developed, including language modeling (Dai & Le 13; Peters *et al.* 53; Radford & Narasimhan 59) and masked language modeling (Dai & Le 13; Peters *et al.* 53); (Kenton & Toutanova 34; Lample & Conneau 37). These techniques significantly improve the performance of natural language processing models (Peters *et al.* 53). Tan & Bansal 71 and Lu *et al.* 42 are two recent publications that expand BERT for visual question answering by pre-training across text-image pairs while masking different picture regions. Chen *et al.* 8 experimented

with converting tables into natural language such that they can be processed by a pre-trained BERT model.

2.6 Table Pre-training

Existing works explore pre-training through several tasks such as Mask Column Prediction in TaBERT [86], Multi-choice Cloze at the Cell Level in TUTA [78], Structure Grounding [16] and SQL execution [41]. My work is closely related to Eisenschlos *et al.* [19], 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 [19], mined from co-occurring tables and NL sentence contexts [86, 30], or directly taken from human-annotated table-NLI datasets [16, 89]. In this study, I employ pre-training data that has been automatically scaled from existing non-NLI data.

This work is particularly based on TAPAS (for Table Parser) [43], which is a weakly supervised question-answering model that reasons over tables without constructing logical forms. TAPAS predicts a minimum programme by identifying a selection of table cells and a potential aggregation operation to be carried out on top of them. Therefore, TAPAS is able to learn operations from natural language without the requirement for formalisation. This is accomplished by expanding the BERT architecture [34] with new embeddings that capture tabular structure and two classification layers for picking cells and predicting a related aggregation operator.

Eisenschlos *et al.* 19 present a pre-training strategy for TAPAS that is essential to its performance on the final job. They extend BERT's masked language model aim to structured data and pre-train the model using millions of tables and relevant Wikipedia text segments. During pre-training, the model masks some tokens from the text segment and the table, with the goal of predicting the original token based on the textual and tabular context.

They propose an end-to-end recipe for differentiable training that enables TAPAS to train with minimal monitoring. For situations involving the selection of only a subset of table cells, they explicitly train the model to identify the gold subset. For situations involving aggregation, the denotation does not reveal the relevant cells or the aggregation procedure. Given the existing model, we compute an anticipated soft scalar result across all aggregation operators and train the model using a regression loss against the gold denotation.

In comparison to previous attempts to reason over tables without generating logical forms, TAPAS achieves higher accuracy and has several advantages: its architecture is simpler as it consists of a single encoder with no auto-regressive decoding; it benefits from pre-training; it handles more question types, including those involving aggregation.

Chapter 3

Data Recasting Framework

This chapter intends to build an understanding of what the problem setting is, and explain the proposed solution for it. It begins by introducing the target task I wish to solve and the base NLI model I build upon. This helps us follow the model pipeline and understand where the resultant augmentation data is supposed to fit in. After this, I go on to propose a framework that transforms non-NLI tabular data into tabular NLI augmentation data.

I describe a generic semi-automatic framework for recasting tabular data for the Table NLI task. By recasting, I mean changing data intended for one job into a format that meets the needs of another task. I begin with analysing the initial facts we require, and identifying them as elements that are either required or desirable. I show why these aspects are exhaustive for the work of recasting and how certain we can be in the truth value of the modified statement, i.e., how confidently we can refer to a resultant perturbed statement as an **Entailment** or **Contradiction**.

This chapter also outlines the challenges and unknowns that affect the framework. I analyse the challenges of identifying table orientations, determining replaceable and irreplaceable table cells and partially matching token-pairs between the tables and the hypothesis.

3.1 Problem Setting

3.1.1 Downstream Task

I utilize TabFact [8], 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. I use the standard train and test splits in our experiments, and report the official accuracy metric. An example from the TabFact dataset is shown in Figure Figure 3.1.

TabFact gives simple and complex tags to each data sample in the train and 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. These are the kind of functions which are unique to inference on tabular data when compared to plain text. Figure 3.2 shows the different kinds

United States House of Representatives Elections, 1972

District	Incumbent	Party	Result	Candidates
California 3	John E. Moss	democratic	re-elected	John E. Moss (d) 69.9% John Rakus (r) 30.1%
California 5	Phillip Burton	democratic	re-elected	Phillip Burton (d) 81.8% Edlo E. Powell (r) 18.2%
California 8	George Paul Miller	democratic	lost renomination democratic hold	Pete Stark (d) 52.9% Lew M. Warden , Jr. (r) 47.1%
California 14	Jerome R. Waldie	republican	re-elected	Jerome R. Waldie (d) 77.6% Floyd E. Sims (r) 22.4%
California 15	John J. Mcfall	republican	re-elected	John J. Mcfall (d) unopposed

Entailed Statement

1. John E. Moss and Phillip Burton are both re-elected in the house of representative election.
2. John J. Mcfall is unopposed during the re-election.
3. There are three different incumbents from democratic.

Refuted Statement

1. John E. Moss and George Paul Miller are both re-elected in the house of representative election.
2. John J. Mcfall failed to be re-elected though being unopposed.
3. There are five candidates in total, two of them are democrats and three of them are republicans.

Figure 3.1: Examples from the TABFACT dataset. The top table contains the semi-structured knowledge facts with caption "United..." . The left and right boxes below provide several entailed and refuted statements. The error parts are highlighted with red font

of complex operations that are included in the TabFact dataset. I report and analyze our results on simple and complex test data separately.

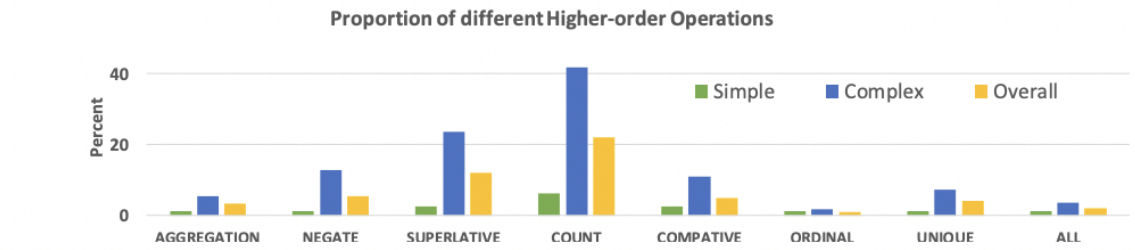


Figure 2: Proportion of different higher-order operations from the simple/complex channels.

Channel	#Sentence	#Table	Len(Ent)	Len(Ref)	Split	#Sentence	Table	Row	Col
Simple	50,244	9,189	13.2	13.1	Train	92,283	13,182	14.1	5.5
Complex	68,031	7,392	14.2	14.2	Val	12,792	1,696	14.0	5.4
Total	118,275	16,573	13.8	13.8	Test	12,779	1,695	14.2	5.4

Figure 3.2: Basic statistics of the data collected from the simple/complex channel and the division of Train/Val/Test Split in the dataset, where "Len" denotes the averaged sentence length.

3.1.2 Model

In all experiments, I start with the Table NLI model developed by Eisenschlos *et al.* [19] as the synthetic data augmentation baseline (referred to as TAPAS + Table-NLI model from here on).

This model architecture is derived from BERT and adds additional embeddings to encode the table structure, following the approach of Herzig *et al.* 30. The statement and table in a pair are tokenized into word pieces and concatenated using the standard [CLS] and [SEP] tokens in between. The table is flattened row by row and no additional separator is added between the cells or rows.

Six types of learnable input embeddings are added together. Token embeddings, position embeddings and segment embeddings are analogous to the ones used in standard BERT. Additionally, this model follows Herzig *et al.* 30 and uses column and row embeddings which encode the two dimensional position of the cell that the token corresponds to and rank embeddings for numeric columns that encode the numeric rank of the cell with respect to the column. This provides a simple way for the model to know how a row is ranked according to a specific column. Bi-directional self-attention mechanism in transformers is unaware of order, which motivates the usage of positional and segment embeddings for text in BERT, and generalizes naturally to column and row embeddings when processing tables, in the 2-dimensional case.

This base model is intermediately pre-trained on automatic rule-based synthetic NLI data to recognize entailment. In this thesis, I follow Eisenschlos *et al.* 19 to add more pre-training steps. These additional pre-training steps are used to expose the model to large amounts of Table NLI data instances which are not part of TabFact. This large scale **augmentation data for pre-training** is what I aim to create through the recasting framework. The output for the recasting framework must be Table NLI data instances, each consisting of a table (premise) and several entailments and/or contradictions (hypotheses) derived from it. The following sections in this chapter explain this framework in detail.

3.2 Prerequisites for Data Recasting

Before I talk about the input for this data recasting framework, I will talk about the resultant output we aim to create. The goal is to create a Table NLI data instance, which consists of

1. A **table**, which is the premise
2. Some **entailments**, i.e. true claims based on the table
3. Some **contradictions**, i.e. false statements based on the table

Original Table (OG)

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

Example #1	
Base Entailment$_{OG}$ (given to us as prerequisite)	Party A won 120 out of 298 seats.
New Entailment$_{OG}$ (substitute entities)	Party B won 89 out of 298 seats.
	***Note that 298, the total value, should remain untouched.*
Paraphrase$_{OG}$ (add linguistic diversity)	Out of a total of 298 available seats, Party B won 89
Contradiction$_{OG}$ (replace entities)	~~Party A~~ Party B won 120 out of 298 seats
Example #2	
Base Entailment$_{OG}$ (given to us as prerequisite)	Party A won the most seats.
New Entailment$_{OG}$ (evaluate ranks and substitute)	Party B won the second most seats.
Paraphrase$_{OG}$ (add linguistic diversity)	Party B secured the second largest number of seats
Contradiction$_{OG}$ (substitute with antonyms)	~~Party A~~ Party B won the most seats.
	Party A won the ~~most~~ least seats.

Table 3.1: Pipeline for generating recasted NLI data. I first create entailments and contradictions from the given base annotation. Apart from the generic process, these two examples show how I deal with superlatives, which entities are irreplaceable and demonstrate creation of contradictions through antonyms.

To be able to generate these, the first and foremost prerequisite is a Table. I choose table based tasks as our source, so the source datasets readily meet this need. In addition to the table, I require at least one reference statement that validates the table. This is a prerequisite because I need the **structure** of this reference statement (henceforth referred to as the **Base Entailment**) to generate further entailments and contradictions. I could alternatively create templates and use their structure to create hypotheses, but those would lead to synthetic data, having limited creativity and versatility. Sourcing structures from

human written sentences adds linguistic diversity and creativity to the resultant dataset. This is one of the major advantages of recasting.

This Base Entailment can be readily available in some datasets. For example, in a generation dataset, the "generated" statements will be Entailments to the table in themselves. In contrast, for Question Answering datasets, I will need to somehow extract a Base Entailment from the given question and its answer. This is explored in further chapters.

Once I have the Base Entailment, the constraints for producing contradictions are rather lax. Falsifying any part of the Base Entailment that is linked to the table creates contradictions. One way of creating contradictions is to negate parts of the statement by adding tokens like "not", "never" and "no". Niven & Kao [48] demonstrate that such modifications encourage the model to learn patterns based on word distributions, and negations are easily correlated with contradictions without the model needing to understand the premise. A similar argument can be made for replacing entities with out-of-table values. Suppose we replace 120 in "Party A won 120 out of 298" with 30 to get "Party A won 30 out of 298". The model can easily recognize that a value like 30, which does not occur in the table at all, is unlikely to belong to entailments. The model hence learns spurious patterns which allows it to classify correctly for the wrong reasons. Hence I aim to create contradictions that have similar word distributions as entailments. One way of doing this is to use values from the table itself to contradict existing statements.

In contrast, to create an entailment, every portion of the perturbed statement must hold true for the entire statement to constitute an entailment. This means that we must be able to verify every portion of the statement. To do this, we must first know "all" the portions of the statement that affect its truth value. Since the truth value of the statement depends only on the table (definition of an entailment is that one should be able to verify it on the basis of the premise alone), we can safely assume that parts of the statement that affect its truth value are those which come from the table. 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 3.1. In the example *"Party A won the **most** seats*, the alignment between *most* and the greatest number of seats must be determined. Although I can employ automatic matching techniques between the Base Entailment and the table to extract relevant entities, I cannot be certain of detecting **all** of them unless they are explicitly provided. Therefore, I must be able to extract the following from source datasets as pre-requisites:

1. A **table** i.e. the Premise
2. A **reference statement** i.e. the Base Entailment - one statement that is true on basis of the table alone
3. **Relevant entities** i.e. entities or values that come directly or indirectly from the table cells

4. **Alignments** of relevant table cells with 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. In the former option, I look at ways of creating new entailments by replacing entities with other candidates in such a way that resultant statement remains true. Since I find alignments between table cells and tokens in the statement, I also keep track of which table cells I use for value substitution. This ensures that our resultant data contains relevant cell information, which can be utilized for supervision. Previous works explore two-step pipelines for table NLI, first identifying the relevant table information and then performing classification with the trimmed premise. Subtask B of SemEval-2021 task 9 [77] requires identification of relevant cells given a tabular premise and hypothesis. Gupta *et al.* [28] argue about models needing to be "right for the right reason". They demonstrate that models often learn spurious correlations and patterns among hypotheses that allow them to classify correctly even with incomplete or noisy premise. It is hence better to not just learn to classify for labels but also learn to extract evidence as an intermediate step. Relevant cell information that I gather in the recasting process can be used for such tasks.

3.3 Perturbing the Hypothesis

In this section, I describe ways to modify the Base Entailment by substituting relevant entities with other potential candidates. I presume that the tables are vertically aligned, which means that the top row contains headers and each column contains entities of the same kind. For simplicity of understanding, I refer to table cells using the coordinate system. A table cell with row coordinate X and column coordinate Y is represented as C_{XY}. A relevant entity in the hypothesis is that which represents some information from the table i.e. an entity which is directly derived from one or more table cells. A *potential candidate* for a relevant entity coming from table cell C_{XY} having coordinates $[rowX, columnY]$ can be any other non-null entity from the same column i.e. $C_{\mathrm{ZY}} | Z \neq X, C_{\mathrm{ZY}} \neq C_{\mathrm{XY}}$.

3.3.1 Creating Entailments

In vertically oriented tables, tables cells in the same row are parts of the same "record". In Table 3.2, our given hypothesis contains relevant entities from 2 different rows. To create entailments, we replace *all* the relevant entities coming from one or more rows in the given Base Entailment with potential candidates. Potential candidates for each entity are shown in Table 3.2. Two or more relevant entities coming from table cells in the same row, say $C_{\mathrm{XA}}, C_{\mathrm{XB}}$, must be substituted with potential candidates from column A and B respectively, such that their row coordinate is equivalent i.e. $C_{\mathrm{XA}}, C_{\mathrm{XB}} \rightarrow C_{\mathrm{ZA}}, C_{\mathrm{ZB}} \mid Z \neq X$ (refer Table 3.1). In the example, I cannot replace Usain Bolt with David Rudisha and 100m with 400m. Even though David Rudisha is a valid potential candidate for replacing Usain Bolt and 400m is a valid potential candidate for replacing 100m. I must replace both entities *Usain Bolt, 100m* simultaneously,

either with *David Rudisha, 800m* or *Wayde van Niekerk, 400m*. Since the other pair of relevant entities *Noah Ngeny, 1000m* come from a different row than *Usain Bolt, 100m*, they should be treated separately. Note that *Noah Ngeny and 1000m* will also be replaced as a pair, since they originate from the same table row and represent a common record. I must also ensure that I do not end up replacing both *Usain Bolt, 100m* and *Noah Ngeny, 1000m* with a common potential candidate such as *Wayde van Niekerk, 400m*. Another important aspect to note is that entities originating from "aggregate rows" (such as the *Total* row in Table 3.1) or "headers" must be left intact, because replacing them would affect the logic of the statement. Table 3.1 shows an example of this.

3.3.2 Creating Contradictions

To create contradictions, I substitute **one or more** relevant entities from the Base Entailment with alternative candidates. Substituting even one relevant entity with contradicting values should falsify the whole statement. In Table 3.2, consider the Base Entailment - *"**Usain Bolt** holds the record in the **100m** event"*. Suppose I replace one entity (**Usain Bolt**) with a potential candidate (Noah Ngeny). I get *"**Noah Ngen** holds the record in the **100m** event"*. This statement is a contradiction. I can choose to replace both relevant entities i.e. **Usain Bolt, 100m** but I will have to ensure that they're not from the same row, which would result in an entailment. Replacing one entity at a time creates contradictions which are minimally differing from their entailment counterparts. Such minimally differing pairs of statements which have opposing labels force the model to learn the relationship between the table and the sentence instead of learning spurious techniques to classify the hypotheses.

However, even while following the above rule, I observe that the ensuing statement may be an entailment by accident. In Table 3.2, consider the Base Entailment - *"**Usain Bolt** holds the record in the **100m** event"*. Suppose I replace only one key entity (**100m**) with a potential candidate (**200m**) to arrive at "*Usain Bolt holds the record in the ~~**100m**~~ **200m*** event". The resultant statement remains an entailment. To prevent this from occurring, the non-replaced entities must be compared. Assume $C_{\text{XA}}, C_{\text{XB}}$ represent the relevant entities in the Base Entailment. If I replace $C_{\text{XA}} \rightarrow C_{\text{ZA}}$ then we must guarantee that $C_{\text{XB}} \neq C_{\text{ZB}}$ to avoid unintentional entailments.

Another conclusion that we can draw from the above rule is that statements containing only one relevant entity can be problematic while creating entailments, because we don't have a supporting entity for it to contradict. For example, consider a simple Base Entailment on Table 3.2 - "**David Rushida** has a world record in his name". No matter what I replace **David Rushida** with, I will get an entailment, because there's no other entity to contradict. As soon as I change the Base Entailment to "**David Rushida** has the **800m** world record in his name", I can easily replace entities to create contradictions. For this reason, I abstain from creating contradictions through substitution for statements have 1 relevant entity. I instead use other techniques as listed below.

Wherever possible, I generate contradictions by substituting antonyms for words in the Base Entailment. This is particularly helpful for scenarios involving superlatives and comparatives. In Table 3.1, Example #2 shows a contradiction being created by replacing "most" with "least". I query NLTK Word-

Net for antonyms and check that the Part of Speech of the antonyms matches that of the original token to maintain coherence in the statement.

Several entailments also have relevant entities which are not directly present in the table, for example results from operations such as counting or aggregation. For example, in Table 3.2, I can have a Base Entailment such as "The athletes on the chart come from **4** distinct nationalities". I replace these numbers with randomly selected numbers in the range of <given number - 10> to <given number + 10>. I abstain taking this route for numbers directly originating from the table to prevent the model from associating out-of-table entities with contradictions. Since both the original and replacement entity in cases of aggregation and counting cannot directly be found in the table, the model cannot get biased to learn spurious relations.

3.3.3 Creating Hypotheses from skeletons

Some source datasets give extensive metadata information about the tabular entities. For example, database-like datasets give information about the data type of entries in each column - string, date, integer, float etc. It is hence possible to identify not just relevant entities but also their data types. I attempt to extract templates from given statements by pulling out entitiy values and any domain or table specific words. For example, suppose I have metadata information for Table 3.2. I know that "Nationality" column has a data type of "text". I can reduce the Base Enatilment "There are 4 distinct nationalities listed on the chart" to "There are <distinct-count-COL1-text> distinct <COL1-text-NAME-plural> listed on the chart". I could now use this skeleton to form sentences from Table 3.1. I could say that since "Party" is a textual column, "There are 3 distinct parties on the chart" is an entailment.

Such skeletons or templates allow me to re-use structures for not just creating statements for a given table, but for any table that satisfies the data-type requirements. This makes it a powerful tool, because instead of writing a fixed set of templates, I can now extract thousands of them from human-written annotations. I have discussed this in detail in chapter 4, where I take dataset specific examples to show how skeletons can be extracted from particular datasets.

Note that there are several restrictions to creating skeletons. Sentences may contain several tokens which are not directly derived from the table, but are not generalised enough to be used everywhere. For example, "Party A won 120 seats" cannot be stripped down to "COL1-text-value won COL2-num-value COL2-num-plural> because "won" is not a general verb that can be used everywhere. To ensure that I only create skeletons which are completely generalisable across domains and tables, I lemmatize the non-functional words in all the skeletons, count their frequencies, and remove the skeletons containing less frequent lemmas. I also perform a degree of manual filtering of the frequency dictionary to ensure no domain specific words end up in the skeletons.

Men's Athletics World Records

Event	Perf	Speed(mph)	Athlete	Nationality	Date
100 m	9.58	23.35	Usain Bolt	Jamaica	16 Aug 2009
200 m	19.19	23.31	Usain Bolt	Jamaica	20 Aug 2009
400 m	43.03	20.79	Wayde van Niekerk	South Africa	14 Aug 2016
800 m	1:40.91	17.734	David Rudisha	Kenya	9 Aug 2012
1000 m	2:11.96	16.952	Noah Ngeny	Kenya	5 Sep 1999
1500 m	3:26.00	16.288	Hicham El Guerrouj	Morocco	14 Jul 1998

Base entailment :

Usain Bolt holds the record for 100m and Noah Ngeny for 1000m.

Potential candidates for Usain Bolt : Wayde van Niekerk, David Rudisha, Hicham El Guerrouj, Noah Ngeny

Potential candidates for 100m : 200m, 400m, 800m, 1000m, 1500m

Potential candidates for Noah Ngeny (Usain Bolt, Wayde van Niekerk, David Rudisha, Hicham El Guerrouj

Potential candidates for 1000m : 100m, 200m, 400m, 800m, 1500m

Potential candidates for (Usain Bolt, 100m) and (Noah Ngeny, 1000m) :

(Usain Bolt, 200m), (Wayde van Niekerk, 400m), (David Rudisha, 800m), (Hicham El Guerrouj, 1500m)

New entailments :

Hicham El Guerrouj holds the record for 1500m and Wayde van Niekerk for 400m .

Usain Bolt holds the record for 200m and David Rudisha for 800m.

Table 3.2: Example of perturbing the base entailment to create new entailments. Table taken from https://en.wikipedia.org/wiki/List_of_world_records_in_athletics

3.4 Perturbing the Table (Premise)

In this subsection, instead of modifying the Base Entailment, I swap two or more **table cells** to **modify the premise** instead of the hypotheses. Similar to Kaushik *et al.* [33] and Gardner *et al.* [20], I build example pairs with minimal differences but opposing inference labels in order to improve model generalisation. These modified tables no longer reflect the real world information. Hence, I 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.* [43] for TabFact.

3.4.1 Creating Counterfactual Tables (CF)

I consider a contradiction C1 formed by replacing the relevant cell $C_{\mathrm{XA}} \rightarrow C_{\mathrm{ZA}}$ in the original table (as described in Table 3.3). To create a counterfactual table, I swap cells $C_{\mathrm{XA}} \leftrightarrow C_{\mathrm{ZA}}$ such that C1 becomes an entailment to the modified table, and the original Base Entailment becomes a contradiction to it. Based on this, I generate further hypotheses, as illustrated in Table 3.3. Note that in Table 3.3, $\mathrm{Contradiction}_{\mathrm{CF}}$ is an $\mathrm{Entailment}_{\mathrm{OG}}$ to the original table, but $\mathrm{Entailment}_{\mathrm{CF}}$ is a $\mathrm{Contradiction}_{\mathrm{OG}}$ to it.

3.4.2 Hypothesis Paraphrasing (HP)

Dagan *et al.* [12] demonstrates that data paraphrasing increases lexical and structural diversity, thus boosting model performance on unstructured NLI. In accordance with Dagan *et al.* [12], I paraphrase our data because the hypotheses derived from Base Entailments have similar structures. For producing paraphrases, I employ the publicly available T5 Model [60] trained on the Google PAWS dataset [91]. I produce the top five paraphrases and then select at random from among them.

Original Table (OG)

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

Counterfactual Table (CF - after cells swaps)

Party Name	Votes(thou)	Seats Won
~~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 3.3: Pipeline for generating counterfactual data taking a contradiction to be the new base annotation. subscript$_{OG}$ represents the "Original" table and subscript$_{CF}$ represents the "Counterfactual" table. Note that in this example, Contradiction$_{CF}$ is an Entailment$_{OG}$ to the original table, but Entailment$_{CF}$ is a Contradiction$_{OG}$ to it.

3.5 Addressing Tabular Recasting Constraints

While this framework provides a generic way of transforming data for TNLI, implementations of it for different datasets bring many challenges to surface. Some challenges are beyond the scope of this work and are counted as limitations. Some challenges are vital to solve, and I address them in the following ways.

3.5.1 Table Orientation.

In the framework described in chapter 3, I have detailed the steps assuming that the tables are vertically aligned. This means that I assume tables to have headers in the top row and have consistent data types in each column. While studying source datasets, I observed several horizontally aligned tables (with the first column containing headers). These would give wrong results if the framework is applied to them. It is hence crucial to identify the alignment of tables.

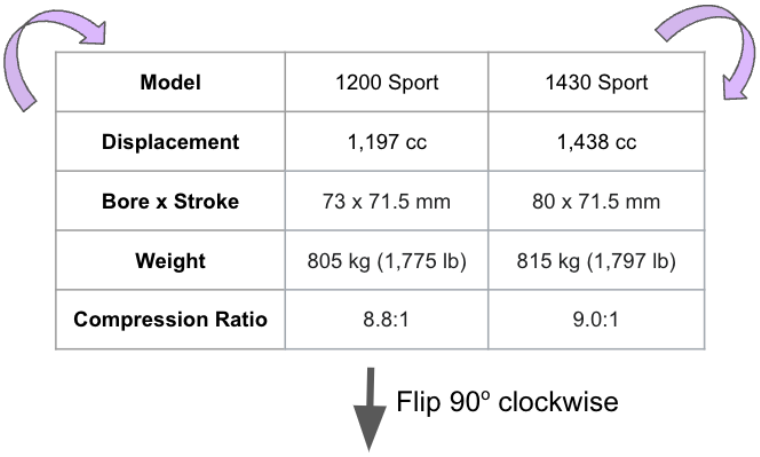


Figure 3.3: Example of a horizontally oriented table and its flipped version.

To deal with this, as a preliminary processing step, I employ heuristics to automatically recognise such tables and subsequently flip them. I mainly use two heuristics for this task. First, assuming that all tables are vertically aligned, I extract their top rows. I create a dictionary of the all header terms like "Name", "Location", "Date", "Nationality" etc occurring in these rows and count their frequencies.

I then choose the top 200 most frequently occurring header terms and look for these terms in the first column of every table. If a table is horizontally aligned, it would have column titles/headers in its first column, which are likely to match with the common header list.

The second heuristic I apply is to check for consistency in data types (numeric, alpha, etc.) across rows rather than columns. I classify the data in each table cell into 4 categories - numeric, alpha, alphanumeric and dates. If the data types seem to both be consistent across rows and inconsistent along columns, the table to likely to be horizontally aligned. Once identified by either heuristic, I flip the table by 90 degrees as shown in the Figure 3.3.

3.5.2 Partial Matching.

I observe that some datasets provide relevant cell information, but do not provide their explicit alignments with the Base Entailment. To fulfill the prerequisites required for recasting, I attempt to match every relevant cell with n-grams in the Base Entailment. If exact string matches are found, I end the search there. However, in many cases, I observe that the statement need not mention the entire table cell entity. This triggers the need for partial matching. Of particular interest is the sample row shown in Table 3.4 that contains names, numbers, locations and dates that are not exact, but partial matches to n-grams in the Base Entailment. People are often referred to from their last name. Similarly, full dates need not be mentioned in cases where year or month is sufficient for the purpose of the statement. Same goes for location - full location may not be mentioned if just the city or country suffices. I attempt to match substrings only in such particular cases, and ensure that the partial substring matches found are not functional words like articles or determiners. I also search for cardinal versions (first, second, third) of ordinals (one, two, three) and numerical values (1,2,3).

3.5.3 Irreplaceable Entities.

I observe that **not all** relevant entities are replaceable by potential candidates. Table 3.3 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 whose substitution will disrupt the truth value of the statement.

Similar observation is made while swapping table cells to create counterfactual tables. Suppose I 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.

US presidential inaugurations (A table row)	
President #	44
Name	Barack Obama
Inauguration Date	January 20, 2009
Location	West Front, United States Capitol

Base Entailment :

Obama's inauguration as the forty fourth president took place at the US Capitol in 2009.

Partial Match	**Case Type**
44 → forty fourth	Ordinal to Cardinal matching
Barack Obama → Obama	Full name to First name/ Last name
January 20, 2009 → 2009	Full date to day/month/year
West Front, United States Capitol → US Capitol	Full location to city/state/country
United States → US	Location Abbreviations to full forms

Table 3.4: An example of cases requiring partial matching.

Chapter 4

Recasting in Practice

4.1 Source Datasets

Using the framework outlined in Chapter 3, I recast five datasets meant for tasks such as Table-to-Text generation, Table Question Answering and Table Semantic Parsing. These datasets are listed in Table 4.1. I choose these tabular tasks because they all perform inference on tables in some manner. Question Answering requires understanding of tables to locate and synthesise the answer. Table-to-Text generation requires creating a descriptive sentence, which would need the model to learn entities in the table and their relations with each other to form a coherent description. Table semantic parsing is the task of converting a given question to its logical form, similar to an SQL query. This would again, require a model to correlate not just textual and tabular cell entries, but also the operations being applied on them.

4.1.1 Choice of Datasets

While the technique of data recasting allows us to utilise data from other datasets, we have to note that this data is likely to have tables sourced from different places, and have different distribution of themes, writing styles and domains than that of the target TabFact test set tables by fuzzy-matching the Table Titles and Source URLs. This ensures that the test data is not accidentally seen by the model, therefore preventing data leakage. There is bound to be some element of domain transfer when different datasets are brought together. To minimise this, I choose datasets that utilise open-domain Wikipedia tables, which is comparable to TabFact. In datasets where distribution of categories and themes is given, I ensure that there are significant overlaps. In addition, these datasets and TabFact share reasoning kinds such as counting, minimum/maximum, ranking, superlatives, comparatives, and uniqueness, among others. 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. I summarize the statistics of the datasets in Table 4.1.

Source Dataset	Task
WikiTableQuestions [51]	Short Form Table Question Answering
FeTaQA [44]	Long Form Table Question Answering
Squall [68]	Tabular Semantic Parsing
WikiSQL [92]	Tabular Semantic Parsing (SQL queries)
ToTTo [49]	Table-to-Text generation

Table 4.1: Source datasets used for creating tabular NLI data

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

Table 4.2: 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.

4.2 Recasting Table2Text Generation datasets

Given a table and a set of highlighted cells, the Table2Text generation task is to create a description derived from the highlighted cells. I presume this description to be the *Base Entailment* given that it is true based on the table. In this case, the highlighted cells become the *relevant entities*. An example is shown in Table 3.3, where *Base Entailment* $_{OG}$ is a description generated from OG Table's highlighted cells.

4.2.1 Recasting ToTTo

ToTTo [49] is a large scale table2text generation dataset. It has over 120k training samples on open-domain Wikipedia tables. ToTTo does not ask annotators to create sentences, it rather searches the Wikipedia page (from where the table was sourced) for sentences containing one or more table cell entities. ToTTo authors then ask annotators to clean these statements, fix issues such as anaphora resolution and grammar, and eliminate any information that cannot be inferred directly from the table. The annotators also mark the highlighted cells. ToTTo states its data quality as "clean".

The advantage of having data that is not directly annotator "generated" but rather annotator "revised" is that it eliminates biases that annotators introduce to a large extent. ToTTo picks sentences from Wikipedia articles. These are freely written articles by humans. Sentences are hence more **natural**,

Table Title: Gabriele Becker
Section Title: International Competitions
Table Description: None

Year	Competition	Venue	Position	Event	Notes
Representing Germany					
1992	World Junior Championships	Seoul, South Korea	10th (semis)	100 m	11.83
1993	European Junior Championships	San Sebastián, Spain	7th	100 m	11.74
			3rd	4x100 m relay	44.60
1994	World Junior Championships	Lisbon, Portugal	12th (semis)	100 m	11.66 (wind: +1.3 m/s)
			2nd	4x100 m relay	44.78
1995	World Championships	Gothenburg, Sweden	7th (q-finals)	100 m	11.54
			3rd	4x100 m relay	43.01

Original Text: After winning the German under-23 100 m title, she was selected to run at the 1995 World Championships in Athletics both individually and in the relay.
Text after Deletion: she at the 1995 World Championships in both individually and in the relay.
Text After Decontextualization: Gabriele Becker competed at the 1995 World Championships in both individually and in the relay.
Final Text: Gabriele Becker competed at the 1995 World Championships both individually and in the relay.

Figure 4.1: An example from the ToTTo dataset, taken from [49].

as compared to task-specific sentences that an annotator might create if asked to generate statements. These sentences are close to the text we might find in the real world, hence making them significant in terms of real life application of the model trained on them.

I treat the human-revised description as the Base Entailment. Since I explicitly know the relevant cells, I make entailments and contradictions through substitution with potential candidates wherever possible. I also create counterfactual tables. I keep the ratio of entailments:contradictions roughly around 1:1. I maintain the train-test split given in the original dataset, i.e. statements derived from ToTTo training samples make up the training set of the recasted data.

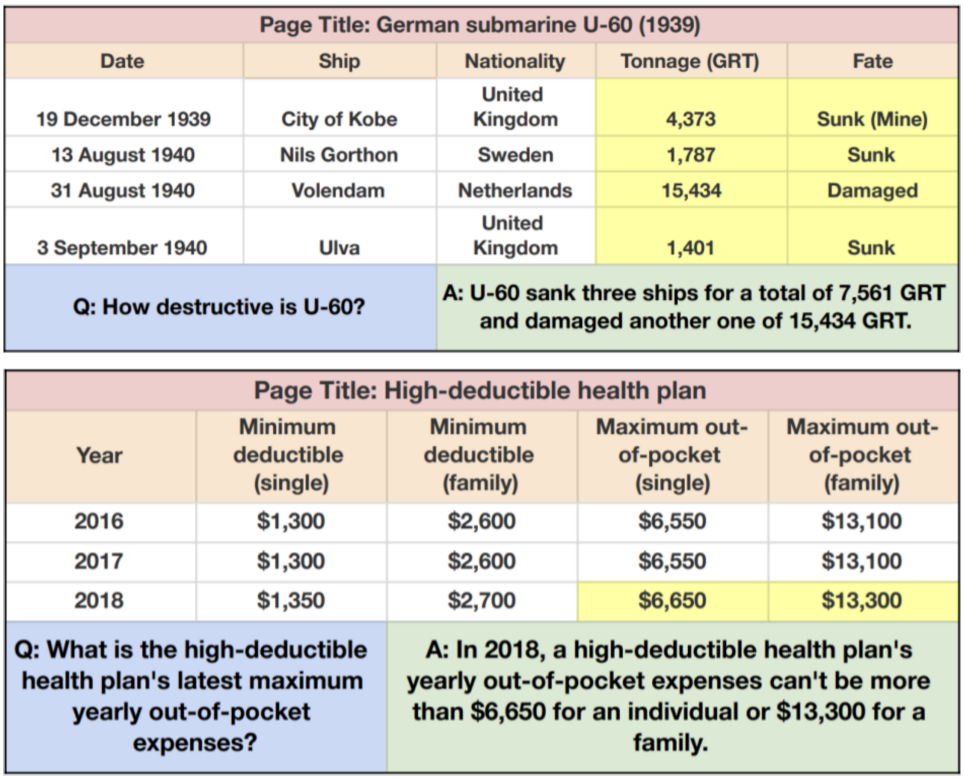
4.3 Recasting Table Question Answering datasets

Table Question Answering is the task where given a table and a question based on the table, one is expected to predict or generate the answer. Answers could be short i.e. one word or one phrase. Answers could also be long-form, i.e. a sentence or a couple of sentences.

Given a question-answering dataset, we know that the information described by the question and its answer is true on the basis of the table. If we can combine the question and answer into a statement, such a statement would entail the table.

4.3.1 Recasting FeTaQA

FeTaQA [44] is a Table Question Answering dataset that was born out of the need for QA datasets to explore complex reasoning. Most existing table question answering datasets prior to FeTaQA contained



Page Title: German submarine U-60 (1939)

Date	Ship	Nationality	Tonnage (GRT)	Fate
19 December 1939	City of Kobe	United Kingdom	4,373	Sunk (Mine)
13 August 1940	Nils Gorthon	Sweden	1,787	Sunk
31 August 1940	Volendam	Netherlands	15,434	Damaged
3 September 1940	Ulva	United Kingdom	1,401	Sunk

Q: How destructive is U-60?

A: U-60 sank three ships for a total of 7,561 GRT and damaged another one of 15,434 GRT.

Page Title: High-deductible health plan

Year	Minimum deductible (single)	Minimum deductible (family)	Maximum out-of-pocket (single)	Maximum out-of-pocket (family)
2016	$1,300	$2,600	$6,550	$13,100
2017	$1,300	$2,600	$6,550	$13,100
2018	$1,350	$2,700	$6,650	$13,300

Q: What is the high-deductible health plan's latest maximum yearly out-of-pocket expenses?

A: In 2018, a high-deductible health plan's yearly out-of-pocket expenses can't be more than $6,650 for an individual or $13,300 for a family.

Figure 4.2: An example from the FeTaQA dataset, taken from [44].

abundant factual questions that primarily evaluate the query and schema comprehension capability of a system, but they failed to include questions that require complex reasoning and integration of information due to the constraint of the associated short-form answers. To address these issues and to demonstrate the full challenge of table question answering, FeTaQA, a long-form Question Answering dataset over tables was introduced. It has 10K Wikipedia-based <table, question, free-form answer, supporting table cells> pairs. FeTaQA yields a more challenging table question answering setting because it requires generating free-form text answers after retrieval, inference, and integration of multiple discontinuous facts from a structured knowledge source. Unlike datasets of generative QA over text in which answers are prevalent with copies of short text spans from the source, answers in the FeTaQA dataset are human-generated explanations involving entities and their high-level relations.

Since FeTaQA provides long-form answers which are statements in themselves, I treat them as entailments. Supporting cell information is given as well, which is helpful for creating contradictions and more entailments by substitution. Even though FeTaQA is a small scale dataset, its training samples are hand-picked to represent complex reasoning, which aligns well with our goal.

I am also able to create counterfactual data from FeTaQA data. Dataset statistics are mentioned in Table 4.2.

Example of FeTaQA Recasting

Ques: 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.

4.3.2 Recasting WikiTableQuestions

WikiTableQuestions [51] is a dataset of 22,033 complex questions on Wikipedia tables, which is made publicly available. It focuses on two important aspects of semantic parsing for question answering, which are the breadth of the knowledge source and the depth of logical compositionality. While most existing work prior to WikiTableQuestions traded off one aspect for another, this dataset was meant for learning to answer complex questions on semi-structured tables using question-answer pairs as supervision. The central challenge here arises from two compounding factors: the broader domain results in an open-ended set of relations, and the deeper compositionality results in a combinatorial explosion in the space of logical forms.

WikiTableQuestions dataset gives <table, question and short-form answer> pairs. Unlike FeTaQA, I cannot directly use these for entailments, but instead, I need to combine the question and short answer to create a logical statement. I explore two ways of doing this - the rule based approach and the neural approach.

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

x_1: *"Greece held its last Summer Olympics in which year?"*
y_1: {2004}

x_2: *"In which city's the first time with at least 20 nations?"*
y_2: {Paris}

x_3: *"Which years have the most participating countries?"*
y_3: {2008, 2012}

x_4: *"How many events were in Athens, Greece?"*
y_4: {2}

x_5: *"How many more participants were there in 1900 than in the first year?"*
y_5: {10}

Figure 4.3: An example from the WikiTableQuestions dataset, taken from [51].

4.3.3 Converting Question-Answer pairs to Statements

Previous research [14] captures the syntactic transformations required to convert a Question Answer pair to a descriptive statement. It is useful for most wh- questions, which makes up a significant portion of our target dataset. I form sentences using the rules derived from this syntactic transformation. While the rules are able to form meaningful statements from some <question, answer> pairs, many input pairs do not fit into the template patterns. I explore the use of neural models to deal with this issue, and to form more fluent and creative statements from questions.

Syntactic transformation for converting a Question-Answer pair to a descriptive sentence

Ques: Where does Jim go to buy groceries? Short answer: Trader Joe's

Where ~~does~~ Jim goes to buy groceries?	remove do-support
~~Where~~ Jim goes where to buy groceries?	reverse wh-movement
Jim goes ~~where~~ to buy groceries ~~?~~	delete question words and mark
Jim goes Farmer Joe's to buy groceries .	plug in the answer
Jim goes to Trader Joe's to buy groceries.	insert preposition

For the neural approach, I use a T5 based pre-trained model developed by Chen *et al.* [5] to convert $\{Question, Answer\} \rightarrow Statement$ (refer Table 1.4). I presume this generated statement to be our Base Entailment. Unless the short-form answer is an aggregate value, it is likely to be an entity from the table. Since it is the "answer" to a question, it is also definitely a relevant entity. I search for full or partial matches between the answer and table cells as well as n-grams in the question. I create contradictions on the basis of any relevant entities found. Note that since all relevant entities are not explicitly given, I do not attempt to create new entailments. Since contradictions can be formed by falsifying any one relevant entity, I allow their creation if matches for relevant entities are found.

Example of WikiTableQuestions Recasting

Ques: What was the total number of seats? Short answer: 298

↓ **combine 298 with the question** ↓

Base statement: There were 298 seats in total.

↓ **combine 298 with the question** ↓

Entailment: There were 298 seats in total. Contradiction: There were 89 seats in total.

4.4 Recasting Semantic Parsing datasets

The task of semantic parsing is to parse a given question to its logical form. Since tables are database-like structured data types, questions can be reduced to SQL-like logical queries, which can be executed

on tables to evaluate answers. In table semantic parsing datasets, we are given a table, a question and its corresponding logical or sql query.

To gather the prerequisites for recasting, I first create databases out of the given csv/json table. I then convert (if required), the given logical form to an executable SQL statement. I execute the statement to get an answer for the corresponding question. Now, similar to Question-Answering datasets, I combine the <Question,Short Answer> pair to create a statement, which is our Base Entailment. Since SQL statements have clear distinction between keywords and values, I can assume the values to be the relevant entities coming from the table. Consider the example "How many females in the chart are over 50 years in age?" and its SQL form:

```
SELECT count(name) FROM table WHERE age > 50 and gender = 'female'
```

I can clearly point out that "name", "age" and "gender" are column headers in this table, and "female", "50 years" are values. I then match the column names and values with the question. If a match is found, I can replace these entities with other potential candidates. The advantage I have with SQL queries is that I can parallelly replace values in questions and SQL queries, and then execute the new query to get an answer. Subsequently, I can combine the updated question and answer to create a new entailment. Since I am executing a logical-form query in the process of creating a new entailment, I can be confident of the label.

I note that it is also interesting to be able to change column names in the SQL query and question parallelly. One thing to be careful about is that I can't just change a column name, I will have to change its corresponding value as well. Applying this to the above given example, it could become -

```
SELECT count(name) FROM table WHERE height > 5 and gender = 'female'
```

Even though I can find both column name "age" and value "50" in the natural language question "How many females in the chart are over 50 years in age?", it would not make sense to replace them with "height" and "5" respectively ("How many females in the chart are over 5 years in height?"). If, suppose, the values were "50 years" and "5 feet" in the table, then the changed sentence would make sense ("How many females in the chart are over 5 feet in height?"). It is, hence, a little tricky to get column-name-changes right.

Our solution for this is to create skeletons as mentioned in chapter 3. Skeletons are completely generalizable statements, because I remove domain-specific words and manually filter them to keep the generic ones.

4.4.1 Recasting WikiSQL

A significant amount of the world's knowledge is stored in relational databases. However, the ability for users to retrieve facts from a database is limited due to a lack of understanding of query languages

Table: CFLDraft

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Question:
How many CFL teams are from York College?

SQL:
```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = "York"
```

Result:
2

Figure 4.4: An example from the WikiSQL dataset, taken from [92].

such as SQL. WikiSQL [92] is a dataset of 80654 hand-annotated examples of questions and SQL queries distributed across 24241 tables from Wikipedia.

To augment the <SQL query, textual question> pair, I parallelly replace values in an SQL query and its corresponding question. I execute the new query, and combine the answer with the perturbed question to create a new entailment.

Example of WikiSQL Recasting

Ques: Which party won 120 seats? SQL: Select party from T where seats = 120

↓ **execute SQL and create statement** ↓

Executed answer: Party A Base Entailment: Party A won 120 seats.

↓ **replace 120 with 89** ↓

Ques': Which party won 89 seats? SQL': Select party from T where seats = 89

↓ **execute SQL' and create statement** ↓

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 I 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 (I find other potential candidates from the answer column). Consider the OG table given in Table 3.3.

4.4.2 Recasting Squall

Large-scale semantic parsing datasets annotated with logical forms have enabled major advances in supervised approaches. Squall [68] was introduced to explore the utility of fine-grained, lexical-level richer supervision. It is a dataset that enriches 11,276 English-language questions from WikiTableQuestions [51] with manually created SQL equivalents plus alignments between SQL and question fragments.

Squall provides token level alignment between not just values and entities, but also aligns SQL keywords with natural language.

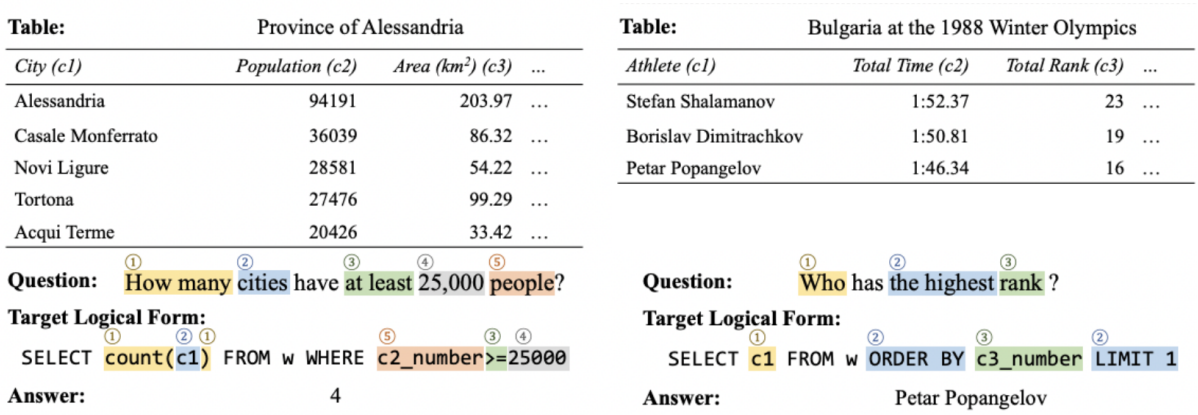


Figure 4.5: An example from the Squall dataset, taken from [68].

I augment Squall similarly 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 3.3 with columns Party (text) and Seats (numeric).

Since Squall gives alignments between SQL keywords and natural language as well, skeletons are richer and almost form a grammar for SQL <-> natural language conversion. For example, the "difference" function in SQL i.e. (A-B) is aligned often with "the difference between A and B".

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?".

4.5 Evaluation and Analysis

In this section I look at evaluating the data I have generated through the recasting framework. I first evaluate the quality and validity of the data through human evaluation. This verifies that the data is indeed coherent and logical. I then run experiments by using this data for pre-training on the downstream TabFact task and present my findings.

4.5.1 Human Evaluation of Recasted Datasets

I create the above 5 datasets with some assumptions in mind, covering as many edge cases as I can. However, it is important to also validate our methods with human evaluation. For this purpose, I asked five annotators to annotate fifty samples from each dataset on two fronts:

- **Inference label**: I ask each annotator to label each sample as entail, refute or neutral. Neutral samples can either be those which can't be derived from the table, or those which don't make sense. This label helps us in identifying how accurate my assumptions are, and how logically correct the generated data is. Some noise is expected, and deemed important for neural models as well, but I should not be creating logically incorrect data.

- **Coherence score**: I ask each annotator to 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 whose meaning can be understood, but the structure or grammar is incorrect in more than one place. Since I claim that our generated data is human-derived, it is important to validate that the fluency and grammar is indeed maintained throughout the data generation process.

I ensure that each sample is annotated by at least 3 annotators, so that a majority can be reached. To consolidate results for human evaluation, I compare our generated label with the majority annotated inference label, and if no majority was reached, I consider the sample inconclusive. For Coherence score of each statement, I calculate the average of the three or more annotators annotating that particular sample. I then average out the scores per dataset.

Analysis. Results are summarized in Table 4.3. I 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 behind at 78%, which is largely due to samples marked as "neutral" or samples where no majority was reached. I 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), I expect our generated sentences to be fluent and semantically correct.

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 4.3: 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.

4.5.2 Experiments on Recasted Datasets

In this section, I examine the relevance of the recasted data across various settings. First, I explain the experimental setup – mainly the pre-training step where augmentation data is introduced. Once I establish that, I present the results in three categories, each aiming to answer one of the following research questions:

1. **RQ1**: *How challenging is recasted data as a Table-NLI benchmark?* I partition the recasted datasets in train and test sets, and evaluate some pre-trained NLI models on recasted test sets.

2. **RQ2**: *How effective are models trained on recasted data in a zero shot setting?* I train Table NLI models only on recasted data and test them on the TabFact simple and complex test sets.

3. **RQ3**: *How beneficial is recasted data for Table NLI data augmentation?* I use recasted data for pre-training NLI models and fine-tune them on TabFact. I present the improvements observed from the base model.

4.5.2.1 Experimental Setup

As described in Chapter 3, I begin with the base model developed by Eisenschlos *et al.* 19. This BERT-based model uses synthetic data for an intermediate pre-training task before it is exposed to the target dataset's training data.

While BERT models for text have been scrutinized and optimized for how to best pre-train and represent textual data, the same attention has not been applied to tabular data, limiting the effectiveness in this setting. The TAPAS-TNLI model [19] addresses these shortcomings using intermediate task pretraining [57], creating efficient data representations, and applying these improvements to the tabular entailment task.

Eisenschlos *et al.* 19 introduces two intermediate pre-training tasks, which are learned from a trained MASK-LM model, one based on synthetic and the other on counterfactual statements. The first one generates a sentence by sampling from a set of logical expressions that filter, combine and compare the

information on the table, which is required in table entailment. The second one corrupts sentences about tables appearing on Wikipedia by swapping entities for plausible alternatives.

Following this approach, I introduce pre-training tasks consisting of our recasted datasets. I treat each dataset as a different pre-training task, since each recasted dataset has different features, and I observe the benefits that each dataset brings to the downstream task.

4.5.2.2 Recasted Data as Evaluation Benchmark

I randomly sample small subsets (typically 10% of the data) from each dataset, including counterfactual tables, to create test sets. I ensure that the test set tables do not overlap with TabFact tables through fuzzy-matching of table URLs and table Titles. I evaluate the publicly available TAPAS-TNLI model [19] fine-tuned on TabFact on the randomly sampled test sets, as shown in Table 4.4. I 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.

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

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.

4.5.2.3 Recasted Data Models in Zero Shot setting

Once I pre-train our model on recasted TNLI data, it is in principle already a table NLI model. Since I create a versatile and large scale dataset, I look at the zero-shot accuracy of our models on the TabFact test set before fine-tuning, as shown in Table 4.5. 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 [18], which is a model trained on synthetic NLI data, by 7% in the zero-shot setting.

Model	TabFact Test Set		
	$\text{Test}_{\text{simple}}$	$\text{Test}_{\text{complex}}$	$\text{Test}_{\text{full}}$
$\text{Table-BERT}_{\text{supervised}}$	79.1	58.2	65.1
$\text{LPA Ranking}_{\text{supervised}}$	78.7	58.5	65.3
$\text{Tapas-RC-Rank}_{\text{zero-shot}}$	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 4.5: 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 [8]. [18] gives the zero-shot accuracy of TAPAS-Row-Col-Rank on TabFact.

Analysis. QA-TNLI achieves the best zero-shot performance of 71.1%. I 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. I 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 on Squall-NLI (i.e. Table Table 4.4) and that 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.

4.5.2.4 Recasted Data for Augmentation

Since TabFact is a binary classification task with Entail and Refute labels, our recasting data can also be used for the purpose of augmentation in this case. I pre-train the model with our recasted data, similar to Eisenschlos *et al.* [19] (refer section 3.1.2), before final fine-tuning on the TabFact dataset. Table 4.6 shows the performance after data augmentation. Our best model outperforms the Table-BERT and LPA Ranking baselines [8] by 17 points, and Eisenschlos *et al.* [19] by 1.1 points.

Analysis. Following the zero-shot results (Table 4.5), QA-TNLI performs well as expected in the fine-tuned setting. I speculate that ToTTo-TNLI outperforms QA-TNLI due to their dataset size disparity (nearly 8x more, refer Table 4.2). The fact that WikiSQL-TNLI achieved the highest accuracy with TabFact-trained models (Table 4.4) and the lowest zero-shot accuracy (Table 4.5) on TabFact

Model	Dev	$\text{Test}_{\text{full}}$	$\text{Test}_{\text{simple}}$	$\text{Test}_{\text{complex}}$	$\text{Test}_{\text{small}}$
Table-BERT-Horizontal[8]	66.1	65.1	79.1	58.2	68.1
LPA-Ranking [8]	65.1	65.3	78.7	58.5	68.9
Logical-Fact-Checker [93]	71.8	71.7	85.4	65.1	74.3
HeterTFV [67]	72.5	72.3	85.9	65.7	74.2
Structure-Aware TF [90]	73.3	73.2	85.5	67.2	-
ProgVGAT [85]	74.9	74.4	88.3	67.6	76.2
TableFormer [83]	82.0	81.6	93.3	75.9	84.6
TAPAS+Salience [76]	82.7	82.1	93.3	76.7	84.3
TAPAS + CF + Syn [19]	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 4.6: Accuracies on TabFact, including the Human Performance. Table-BERT-Horizontal and LPA-Ranking (w/ discriminator) are baselines taken from TabFact [8]. 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 [19] 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.

indicates that the data is relatively non-complex. Squall-TNLI does not improve model performance after augmention despite its remarkable zero-shot performance (Table 4.4). I suspect that this is because the domains and types of underlying logic (a.k.a. reasoning types) are quite distinct. I 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. I 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.

4.5.3 Combinations

In an effort to see if data volume is directly proportional to model performance, I combined all datasets to train a model. I mixed samples from all datasets in equal parts for each training epoch.

The intuition was that mixing such different data might not be extremely beneficial, especially since the distribution of categories, aggregation types, language etc is vastly different across datasets. The results reflect the same. However, I do note that I mixed the two Question Answering datasets, which yielded positive results. Perhaps this is because there is some commonality in the type of the data these datasets comprise.

Chapter 5

Conclusion, Limitations and Future Work

In this work I introduced a semi-automatic framework for recasting tabular data. I made the case for choosing the recasting route due to its cost effectiveness, scalability and ability to retain human-like diversity in the resultant data. I proposed a framework to recast existing tabular datasets for the task of Natural Language Inference. I then leveraged this framework to generate NLI data for five existing tabular datasets. In addition, I demonstrated that the recasted datasets could be utilized as evaluation benchmarks as well as for data augmentation to enhance performance on the Tabular NLI task presented by TabFact [8]. I also showed its utility as an evaluation benchmark and as training data in a zero-shot setting.

5.1 Limitations

This work on recasting tabular data yields some interesting outcomes. However, I note that there are some limitations to this direction of work, which are inevitably introduced in the process. These limitations pave the way for future work, as well as show room for improvement and further investigation.

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, I 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 [46] 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 5.1, 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.

Micheal Phelps - 100m Butterfly

Venue	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 5.1: An example table and an entailment derived from the same.

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.

5.2 Future Work

My work explores a unidirectional framework for recasting data from X to Tabular NLI. A natural extension to this line of work would be to investigate the other way around, to see if NLI data can be used for tasks such as Question Answering, Generation and Semantic Parsing. To exhaustively cover the benefits of recasting in a tabular setting, such investigation is the most evident next step. The foundation

for the problem has already been laid and similar domain datasets from different research tasks have already been established.

Future work in this direction can also lead to an interesting parallel dataset being formed for multiple tabular tasks. I have already produced parallel data for two tasks at a time (NLI and x), but it would be highly beneficial to have parallel data in more than 2 tasks. This would entail that we can train models which are consistent over all tasks. For example, a model will learn that if a question Q has an answer A then the statement combining Q and A is an entailment and vice versa. Models can also be trained to perform intermediate tasks when training for a target task and use the outputs from the "intermediate" tasks to boost their confidence. For example, if the model can predict the answer A to the question Q in an intermediate task, its confidence in ruling statement S as an entailment in the downstream NLI task would be much higher. Having parallel data can create opportunities for such models.

A different direction of work can focus on improving the data augmentation setting of my work. This work describes a framework to produce large scale data, but all data may not always be useful data. There is ample scope in investigating methods of choosing samples for augmentation and training, which can significantly improve results. It would also help make a more challenging and comprehensive test set.

Along similar lines, there is also scope for investigating how combinations of recasted datasets influence model performance. I did perform some experiments with combinations of datasets, but there are several different training methods that can be used when mixing data from different sources. The ratio and order in which each kind of data is shown to the model can make a difference. This line of work can be investigated independently of recasting as well.

In conclusion, my work explores the themes of data recasting, data augmentation, tabular inference and model pre-training. This work, along with its limitations and scope for future work, is an important advancement in the field of tabular reasoning, which is a fairly new and data-scarce domain.

Related Publications

1. **Aashna Jena**, Vivek Gupta, Julian Martin Eisenschlos and Manish Shrivastava. Leveraging Data Recasting to Enhance Tabular Inference. *Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4512 - 4525* (https://aclanthology.org/2022.findings-emnlp.328/)

2. **Aashna Jena**, Vivek Gupta, Julian Martin Eisenschlos and Manish Shrivastava. Framework for Recasting Table-to-Text Generation Data for Tabular Inference. *Structured and Unstructured Knowledge Integration Workshop at Annual Conference of the North American Chapter of the Association for Computational Linguistics 2022* (https://suki-workshop.github.io/assets/paper/26.pdf)

Bibliography

[1] Abbas, Faheem, Malik, Muhammad Kamran, Rashid, Muhammad Umair, & Zafar, Rizwan. 2016. WikiQA—A question answering system on Wikipedia using freebase, DBpedia and Infobox. *Pages 185–193 of: 2016 Sixth International Conference on Innovative Computing Technology (INTECH)*. IEEE.

[2] Alberti, Chris, Andor, Daniel, Pitler, Emily, Devlin, Jacob, & Collins, Michael. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. *Pages 6168–6173 of: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

[3] Aly, Rami, Guo, Zhijiang, Schlichtkrull, Michael Sejr, Thorne, James, Vlachos, Andreas, Christodoulopoulos, Christos, Cocarascu, Oana, & Mittal, Arpit. 2021. The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task. *Pages 1–13 of: Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Dominican Republic: Association for Computational Linguistics.

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

[5] Chen, Jifan, Choi, Eunsol, & Durrett, Greg. 2021a. Can NLI Models Verify QA Systems' Predictions? *Pages 3841–3854 of: Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics.

[6] Chen, Wenhu, Zha, Hanwen, Chen, Zhiyu, Xiong, Wenhan, Wang, Hong, & Wang, William Yang. 2020a. HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data. Nov., 1026–1036.

[7] Chen, Wenhu, Chen, Jianshu, Su, Yu, Chen, Zhiyu, & Wang, William Yang. 2020b. Logical Natural Language Generation from Open-Domain Tables. *Pages 7929–7942 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

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

[9] Chen, Zhiyu, Chen, Wenhu, Zha, Hanwen, Zhou, Xiyou, Zhang, Yunkai, Sundaresan, Sairam, & Wang, William Yang. 2020d. Logic2Text: High-Fidelity Natural Language Generation from Logical Forms. *Pages 2096–2111 of: Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics.

[10] Chen, Zhiyu, Chen, Wenhu, Smiley, Charese, Shah, Sameena, Borova, Iana, Langdon, Dylan, Moussa, Reema, Beane, Matt, Huang, Ting-Hao, Routledge, Bryan, & Wang, William Yang. 2021b. FinQA: A Dataset of Numerical Reasoning over Financial Data. *Pages 3697–3711 of: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

[11] Dagan, Ido, Glickman, Oren, & Magnini, Bernardo. 2007. The PASCAL Recognising Textual Entailment Challenge. *In: Machine Learning Challenges Workshop*.

[12] Dagan, Ido, Roth, Dan, Sammons, Mark, & Zanzotto, Fabio Massimo. 2013. Recognizing Textual Entailment: Models and Applications. *Synthesis Lectures on Human Language Technologies*, **6**(07), 1–220.

[13] Dai, Andrew M., & Le, Quoc V. 2015. Semi-supervised Sequence Learning. *In: NIPS*.

[14] Demszky, Dorottya, Guu, Kelvin, & Liang, Percy. 2018. Transforming Question Answering Datasets Into Natural Language Inference Datasets. *ArXiv*, **abs/1809.02922**.

[15] Deng, Li, Zhang, Shuo, & Balog, Krisztian. 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*.

[16] Deng, Xiang, Awadallah, Ahmed Hassan, Meek, Christopher, Polozov, Oleksandr, Sun, Huan, & Richardson, Matthew. 2021. Structure-Grounded Pretraining for Text-to-SQL. *Pages 1337–1350 of: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics.

[17] Dong, Li, & Lapata, Mirella. 2016. Language to Logical Form with Neural Attention. *Pages 33–43 of: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics.

[18] Dong, Rui, & Smith, David. 2021. Structural Encoding and Pre-training Matter: Adapting BERT for Table-Based Fact Verification. *Pages 2366–2375 of: Proceedings of the 16th Conference of*

the European Chapter of the Association for Computational Linguistics: Main Volume. Online: Association for Computational Linguistics.

[19] Eisenschlos, Julian, Krichene, Syrine, & Müller, Thomas. 2020. Understanding tables with intermediate pre-training. *Pages 281–296 of: Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics.

[20] Gardner, Matt, Artzi, Yoav, Basmov, Victoria, Berant, Jonathan, Bogin, Ben, Chen, Sihao, Dasigi, Pradeep, Dua, Dheeru, Elazar, Yanai, Gottumukkala, Ananth, Gupta, Nitish, Hajishirzi, Hannaneh, Ilharco, Gabriel, Khashabi, Daniel, Lin, Kevin, Liu, Jiangming, Liu, Nelson F., Mulcaire, Phoebe, Ning, Qiang, Singh, Sameer, Smith, Noah A., Subramanian, Sanjay, Tsarfaty, Reut, Wallace, Eric, Zhang, Ally, & Zhou, Ben. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. *Pages 1307–1323 of: Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics.

[21] Geiger, Atticus, Cases, Ignacio, Karttunen, Lauri, & Potts, Christopher. 2019. Posing Fair Generalization Tasks for Natural Language Inference. *Pages 4485–4495 of: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

[22] Geva, Mor, Goldberg, Yoav, & Berant, Jonathan. 2019. Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. *Pages 1161–1166 of: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

[23] Geva, Mor, Gupta, Ankit, & Berant, Jonathan. 2020. Injecting Numerical Reasoning Skills into Language Models. *Pages 946–958 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[24] Ghasemi-Gol, Majid, & Szekely, Pedro A. 2018. TabVec: Table Vectors for Classification of Web Tables. *ArXiv*, **abs/1802.06290**.

[25] Glass, Michael, Canim, Mustafa, Gliozzo, Alfio, Chemmengath, Saneem, Kumar, Vishwajeet, Chakravarti, Rishav, Sil, Avi, Pan, Feifei, Bharadwaj, Samarth, & Fauceglia, Nicolas Rodolfo. 2021. Capturing Row and Column Semantics in Transformer Based Question Answering over Tables. *Pages 1212–1224 of: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics.

[26] Gupta, Vivek, Mehta, Maitrey, Nokhiz, Pegah, & Srikumar, Vivek. 2020. INFOTABS: Inference on Tables as Semi-structured Data. *Pages 2309–2324 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[27] Gupta, Vivek, Bhat, Riyaz A., Ghosal, Atreya, Srivastava, Manish, Singh, Maneesh, & Srikumar, Vivek. 2021. Is My Model Using The Right Evidence? Systematic Probes for Examining Evidence-Based Tabular Reasoning. *CoRR*, **abs/2108.00578**.

[28] Gupta, Vivek, Zhang, Shuo, Vempala, Alakananda, He, Yujie, Choji, Temma, & Srikumar, Vivek. 2022. Right for the Right Reason: Evidence Extraction for Trustworthy Tabular Reasoning. *Pages 3268–3283 of: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics.

[29] Gururangan, Suchin, Swayamdipta, Swabha, Levy, Omer, Schwartz, Roy, Bowman, Samuel, & Smith, Noah A. 2018. Annotation Artifacts in Natural Language Inference Data. *Pages 107–112 of: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics.

[30] Herzig, Jonathan, Nowak, Pawel Krzysztof, Müller, Thomas, Piccinno, Francesco, & Eisenschlos, Julian. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. *Pages 4320–4333 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[31] Iida, Hiroshi, Thai, Dung, Manjunatha, Varun, & Iyyer, Mohit. 2021. TABBIE: Pretrained Representations of Tabular Data. *Pages 3446–3456 of: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics.

[32] Jia, Robin, & Liang, Percy. 2016. Data Recombination for Neural Semantic Parsing. *Pages 12–22 of: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics.

[33] Kaushik, Divyansh, Hovy, Eduard, & Lipton, Zachary. 2020. Learning The Difference That Makes A Difference With Counterfactually-Augmented Data. *In: International Conference on Learning Representations*.

[34] Kenton, Jacob Devlin Ming-Wei Chang, & Toutanova, Lee Kristina. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Pages 4171–4186 of: Proceedings of NAACL-HLT*.

[35] Khashabi, Daniel, Chaturvedi, Snigdha, Roth, Michael, Upadhyay, Shyam, & Roth, Dan. 2018. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences.

Pages 252–262 of: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics.

[36] Khot, Tushar, Sabharwal, Ashish, & Clark, Peter. 2018. SciTaiL: A Textual Entailment Dataset from Science Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**(1).

[37] Lample, Guillaume, & Conneau, Alexis. 2019. Cross-lingual Language Model Pretraining. *In: Neural Information Processing Systems*.

[38] Leonandya, Rezka, Hupkes, Dieuwke, Bruni, Elia, & Kruszewski, Germán. 2019. The Fast and the Flexible: Training Neural Networks to Learn to Follow Instructions from Small Data. *Pages 223–234 of: Proceedings of the 13th International Conference on Computational Semantics - Long Papers*. Gothenburg, Sweden: Association for Computational Linguistics.

[39] Levesque, Hector, Davis, Ernest, & Morgenstern, Leora. 2012. The winograd schema challenge. *In: Thirteenth international conference on the principles of knowledge representation and reasoning*.

[40] Lewis, Patrick, Denoyer, Ludovic, & Riedel, Sebastian. 2019. Unsupervised Question Answering by Cloze Translation. *Pages 4896–4910 of: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

[41] Liu, Qian, Chen, Bei, Guo, Jiaqi, Ziyadi, Morteza, Lin, Zeqi, Chen, Weizhu, & guang Lou, Jian. 2021. *TAPEX: Table Pre-training via Learning a Neural SQL Executor*.

[42] Lu, Jiasen, Batra, Dhruv, Parikh, Devi, & Lee, Stefan. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *In: Neural Information Processing Systems*.

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

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

[45] Neelakantan, Arvind, Roth, Benjamin, & McCallum, Andrew. 2015. Compositional Vector Space Models for Knowledge Base Completion. *Pages 156–166 of: 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). Beijing, China: Association for Computational Linguistics.

[46] Neeraja, J., Gupta, Vivek, & Srikumar, Vivek. 2021. Incorporating External Knowledge to Enhance Tabular Reasoning. *In: Proceedings of the 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[47] Nie, Yixin, Williams, Adina, Dinan, Emily, Bansal, Mohit, Weston, Jason, & Kiela, Douwe. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. *Pages 4885–4901 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[48] Niven, Timothy, & Kao, Hung-Yu. 2019. Probing Neural Network Comprehension of Natural Language Arguments. *Pages 4658–4664 of: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

[49] Parikh, Ankur, Wang, Xuezhi, Gehrmann, Sebastian, Faruqui, Manaal, Dhingra, Bhuwan, Yang, Diyi, & Das, Dipanjan. 2020. ToTTo: A Controlled Table-To-Text Generation Dataset. *Pages 1173–1186 of: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

[50] Parmar, Mihir, Mishra, Swaroop, Geva, Mor, & Baral, Chitta. 2022. Don't Blame the Annotator: Bias Already Starts in the Annotation Instructions. *arXiv preprint arXiv:2205.00415*.

[51] Pasupat, Panupong, & Liang, Percy. 2015a. Compositional Semantic Parsing on Semi-Structured Tables. *Pages 1470–1480 of: 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)*.

[52] Pasupat, Panupong, & Liang, Percy. 2015b. Compositional Semantic Parsing on Semi-Structured Tables. *Pages 1470–1480 of: 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)*. Beijing, China: Association for Computational Linguistics.

[53] Peters, Matthew E., Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, & Zettlemoyer, Luke. 2018. Deep Contextualized Word Representations. *Pages 2227–2237 of: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics.

[54] Poliak, Adam, Naradowsky, Jason, Haldar, Aparajita, Rudinger, Rachel, & Van Durme, Benjamin. 2018a. Hypothesis Only Baselines in Natural Language Inference. June, 180–191.

[55] Poliak, Adam, Haldar, Aparajita, Rudinger, Rachel, Hu, J. Edward, Pavlick, Ellie, White, Aaron Steven, & Durme, Benjamin Van. 2018b. Towards a Unified Natural Language Inference Framework to Evaluate Sentence Representations. *CoRR*, **abs/1804.08207**.

[56] Pramanick, Aniket, & Bhattacharya, Indrajit. 2021. Joint Learning of Representations for Webtables, Entities and Types using Graph Convolutional Network. *Pages 1197–1206 of: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics.

[57] Pruksachatkun, Yada, Phang, Jason, Liu, Haokun, Htut, Phu Mon, Zhang, Xiaoyi, Pang, Richard Yuanzhe, Vania, Clara, Kann, Katharina, & Bowman, Samuel R. 2020. Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work? *Pages 5231–5247 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[58] Qin, Lianhui, Shwartz, Vered, West, Peter, Bhagavatula, Chandra, Hwang, Jena D., Le Bras, Ronan, Bosselut, Antoine, & Choi, Yejin. 2020. Back to the Future: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning. *Pages 794–805 of: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

[59] Radford, Alec, & Narasimhan, Karthik. 2018. Improving Language Understanding by Generative Pre-Training.

[60] Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, & Liu, Peter J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, **21**(140), 1–67.

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

[62] Ran, Qiu, Lin, Yankai, Li, Peng, Zhou, Jie, & Liu, Zhiyuan. 2019. NumNet: Machine Reading Comprehension with Numerical Reasoning. *Pages 2474–2484 of: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

[63] Ravichander, Abhilasha, Naik, Aakanksha, Rose, Carolyn, & Hovy, Eduard. 2019. EQUATE: A Benchmark Evaluation Framework for Quantitative Reasoning in Natural Language Inference. *Pages 349–361 of: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics.

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

[65] Sammons, Mark, Vydiswaran, V.G.Vinod, & Roth, Dan. 2010. "Ask Not What Textual Entailment Can Do for You...". *Pages 1199–1208 of: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics.

[66] Sellam, Thibault, Das, Dipanjan, & Parikh, Ankur. 2020. BLEURT: Learning Robust Metrics for Text Generation. *Pages 7881–7892 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[67] Shi, Qi, Zhang, Yu, Yin, Qingyu, & Liu, Ting. 2020a. Learn to Combine Linguistic and Symbolic Information for Table-based Fact Verification. *Pages 5335–5346 of: Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics.

[68] Shi, Tianze, Zhao, Chen, Boyd-Graber, Jordan L., Daum'e, Hal, & Lee, Lillian. 2020b. On the Potential of Lexico-logical Alignments for Semantic Parsing to SQL Queries. *In: FINDINGS*.

[69] Suhr, Alane, Lewis, Mike, Yeh, James, & Artzi, Yoav. 2017. A Corpus of Natural Language for Visual Reasoning. *Pages 217–223 of: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics.

[70] Talmor, Alon, Herzig, Jonathan, Lourie, Nicholas, & Berant, Jonathan. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. *Pages 4149–4158 of: 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)*. Minneapolis, Minnesota: Association for Computational Linguistics.

[71] Tan, Hao Hao, & Bansal, Mohit. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *ArXiv*, **abs/1908.07490**.

[72] Trabelsi, Mohamed, Davison, Brian D., & Heflin, Jeff. 2019. Improved Table Retrieval Using Multiple Context Embeddings for Attributes. *Pages 1238–1244 of: 2019 IEEE International Conference on Big Data (Big Data)*.

[73] Trivedi, Harsh, Kwon, Heeyoung, Khot, Tushar, Sabharwal, Ashish, & Balasubramanian, Niranjan. 2019. Repurposing Entailment for Multi-Hop Question Answering Tasks. *Pages 2948–2958 of:*

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). Minneapolis, Minnesota: Association for Computational Linguistics.

[74] Vlachos, Andreas, & Riedel, Sebastian. 2015. Identification and Verification of Simple Claims about Statistical Properties. *Pages 2596–2601 of: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics.

[75] Wallace, Eric, Wang, Yizhong, Li, Sujian, Singh, Sameer, & Gardner, Matt. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. *Pages 5307–5315 of: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

[76] Wang, Fei, Sun, Kexuan, Pujara, Jay, Szekely, Pedro, & Chen, Muhao. 2021a. Table-based Fact Verification With Salience-aware Learning. *Pages 4025–4036 of: Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics.

[77] Wang, Nancy X. R., Mahajan, Diwakar, Danilevsky, Marina, & Rosenthal, Sara. 2021b. SemEval-2021 Task 9: Fact Verification and Evidence Finding for Tabular Data in Scientific Documents (SEM-TAB-FACTS). *Pages 317–326 of: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Online: Association for Computational Linguistics.

[78] Wang, Zhiruo, Dong, Haoyu, Jia, Ran, Li, Jia, Fu, Zhiyi, Han, Shi, & Zhang, Dongmei. 2021c. TUTA: Tree-Based Transformers for Generally Structured Table Pre-Training. *Page 1780–1790 of: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and; Data Mining*. KDD '21. New York, NY, USA: Association for Computing Machinery.

[79] White, Aaron Steven, Rastogi, Pushpendre, Duh, Kevin, & Van Durme, Benjamin. 2017. Inference is Everything: Recasting Semantic Resources into a Unified Evaluation Framework. *Pages 996–1005 of: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing.

[80] Williams, Adina, Nangia, Nikita, & Bowman, Samuel. 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*.

[81] Wu, Changxing, Shi, Xiaodong, Chen, Yidong, Huang, Yanzhou, & Su, Jinsong. 2016. Bilingually-constrained Synthetic Data for Implicit Discourse Relation Recognition. *Pages 2306–2312 of: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics.

[82] Xiong, Wenhan, Du, Jingfei, Wang, William Yang, & Stoyanov, Veselin. 2020. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. *In: ICLR*.

[83] Yang, Jingfeng, Gupta, Aditya, Upadhyay, Shyam, He, Luheng, Goel, Rahul, & Paul, Shachi. 2022. TableFormer: Robust Transformer Modeling for Table-Text Encoding. *In: ACL*.

[84] Yang, Xiaoyu, & Zhu, Xiaodan. 2021. Exploring Decomposition for Table-based Fact Verification. *Pages 1045–1052 of: Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics.

[85] Yang, Xiaoyu, Nie, Feng, Feng, Yufei, Liu, Quan, Chen, Zhigang, & Zhu, Xiaodan. 2020. Program Enhanced Fact Verification with Verbalization and Graph Attention Network. *Pages 7810–7825 of: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

[86] Yin, Pengcheng, Neubig, Graham, Yih, Wen-tau, & Riedel, Sebastian. 2020a. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. *Pages 8413–8426 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[87] Yin, Pengcheng, Neubig, Graham, Yih, Wen-tau, & Riedel, Sebastian. 2020b. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. *Pages 8413–8426 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[88] Yoran, Ori, Talmor, Alon, & Berant, Jonathan. 2021. Turning Tables: Generating Examples from Semi-structured Tables for Endowing Language Models with Reasoning Skills. *arXiv preprint arXiv:2107.07261. Version 1.*

[89] Yu, Tao, Wu, Chien-Sheng, Lin, Xi Victoria, Wang, Bailin, Tan, Yi Chern, Yang, Xinyi, Radev, Dragomir R., Socher, Richard, & Xiong, Caiming. 2021. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. *In: International Conference of Learning Representation*.

[90] Zhang, Hongzhi, Wang, Yingyao, Wang, Sirui, Cao, Xuezhi, Zhang, Fuzheng, & Wang, Zhongyuan. 2020. Table Fact Verification with Structure-Aware Transformer. *Pages 1624–1629 of: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

[91] Zhang, Yuan, Baldridge, Jason, & He, Luheng. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. *Pages 1298–1308 of: 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)*. Minneapolis, Minnesota: Association for Computational Linguistics.

[92] Zhong, Victor, Xiong, Caiming, & Socher, R. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *ArXiv*, **abs/1709.00103**.

[93] Zhong, Wanjun, Tang, Duyu, Feng, Zhangyin, Duan, Nan, Zhou, Ming, Gong, Ming, Shou, Linjun, Jiang, Daxin, Wang, Jiahai, & Yin, Jian. 2020. LogicalFactChecker: Leveraging Logical Operations for Fact Checking with Graph Module Network. *Pages 6053–6065 of: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

[94] Zhou, Ben, Khashabi, Daniel, Ning, Qiang, & Roth, Dan. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. *Pages 3363–3369 of: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.