Efficient Realistic Data Generation Framework for Semi-Structured Tabular Inference

Dibyakanti Kumar[1*], **Vivek Gupta**[2*†], **Soumya Sharma**[1]
[1]IIT Guwahati; [2]University of Utah; [3]IIT Kharagpur;
dibyakan@iitg.ac.in; vgupta@cs.utah.edu; soumyasharma20@gmail.com

Abstract

Traditional approaches for creating natural language inference datasets from semi-structured datasets either involve manually constructing datasets, which is a computationally expensive and time-consuming operation which limits its generation to small-scale datasets, or use fully automated methods, which while capable of producing large-scale datasets frequently produce basic statements that may lack reasoning. In this paper, we produce a semi-automated framework for data generation from entity based tabular data. We use the INFOTABS (Gupta et al., 2020) dataset to generate a large-scale human-like synthetic data that includes counterfactual entity-based tables. Through thorough experiments we demonstrate the performance of the generated dataset.

1 Introduction

Natural Language Inference (NLI) is a Natural Language Processing task of determining if a hypothesis is entailed or contradicted given a premise, or is unrelated to it (Dagan et al., 2013). This task was extended to include tabular premises in TABFACT (Chen et al., 2020b) and INFOTABS (Gupta et al., 2020) datasets. To encode tabular data into a form amenable for transformer based models, they are flattened into artificial sentences using heuristics (Chen et al., 2020b; Gupta et al., 2020; Eisenschlos et al., 2020; Yin et al., 2020a, and others).

While manually generated datasets often are biased and require extensive manual efforts in terms of time and money, automatically generated datasets can be simple and lack reasoning. In this paper, we improve on the dataset creation processes shown in Müller et al. (2021), TABFACT (Chen et al., 2020b) and INFOTABS (Gupta et al., 2020) and create a framework to ease the automatic generation of sentences from tabular data. Through a semi-automated framework we can limit the human effort to creating a set of key-specific rules-based templates based on the category of a table. These templates can then be used as a fill-in-the-blanks based on the actual tabular data values.

We envision this framework to help generate large-scale datasets that could potentially serve as an augmentation dataset or when the training data is limited. In this paper we make the following contributions:

1. We create a semi-automatic framework for

Janet Leigh (Original)			Janet Leigh (Counter-Factual)		
Born	July 6, 1927		**Born**	July 6, 1927	
Died	October 3, 2004		**Died**	January 13, 1994	
Children	Kelly Curtis; Jamie Lee Curtis		**Children**	Kelly Curtis	
Alma Mater	University of the Pacific		**Alma Mater**	University of California	
Occupation	None		**Occupation**	Scientist	
H1: Janet Leigh was born before 1940.		E	H1: Janet Leigh was born after 1915.		E
H2: The age of Janet Leigh is more than 70.		E	H2: The age of Janet Leigh is more than 70.		C
H3: Janet Leigh has 1 children		C	H3: Janet Leigh has more than 2 children.		C
H4: Janet Leigh graduated from University of the Pacific.		E	H4: Janet Leigh graduated from University of the Pacific.		C

Table 1: An example of original and counter-factual table from the category Person. Here we have showcased how different keys can be modified using multiple operations. We have also shown how the labels (E - Entailment, C - Contradictory) for a particular key specific hypothesis may change. In the example table of "Janet Leigh" the first column represent the keys(e.g. Born; Died etc) and the second column represent the corresponding values (e.g. July 6,1927; October 3, 2004 etc).

[*] Equal Contribution
[†] Corresponding Author

data generation from entity based tabular data. The framework gives the ability to generate human-alike large-scale data with minimal human intervention.

2. We use the framework to extend the INFOTABS (Gupta et al., 2020) dataset and create a large-scale human-like synthetic data AUTO-TNLI that also contains counterfactual entity-based tables.

3. We demonstrate the AUTO-TNLI complexity and the benefits of utilizing it as an augmentation dataset through experiments using RoBERTa (Liu et al., 2019) model (~1.9-2% performance improvement). Furthermore, we also explore semi-supervised limited training setting, and observe a 12.22% improvement over just INFOTABS finetuning with augmentation.

The dataset and associated scripts, is available at `https://autotnli.github.io`.

2 Proposed Framework

This section describes the semi-automatic framework for generating human-like inferential hypothesis sentences (i.e. true/false) from semi-structured tabular data as premise, more specifically entity-table data such as Wikipedia InfoBoxes.

Müller et al. (2021) demonstrated that adding counterfactual hypothesis sentences enhances model performance on the TABFACT tabular inference dataset. Dagan et al. (2013) demonstrated that inserting paraphrases boost lexical diversity of premise and thus enhanced model performance on unstructured NLI. Chen et al. (2020a) demonstrated in NLG how automatic frameworks lack the ability generate logical sentences with non-superficial reasoning. We draw on these findings to construct AUTO-TNLI which incorporate counterfactual tables created using original INFOTABS tables, as well as paraphrase premise statements.

Next, we would describe the main component of our proposed framework (a.) Hypothesis Template Creation, (b.) Rational Counterfactual Table Creation, (c.) Paraphrasing of Premise Tables, and (d.) Automatic Table-Hypothesis Generation.

2.1 Hypothesis Template Creation

For a particular category of a table (E.g. *Movie*), the attributes are consistent across all tables (e.g. *Length*, *Producer*, *Director*, and others). Therefore, one can write **key-based rules** to create logical hypothesis sentences. We created such key-based rules for the following reasoning types : (a.) Temporal Reasoning, (b.) , Numerical Reasoning, (c.) Spatial Reasoning, (d.) Common Sense Reasoning. Ta-

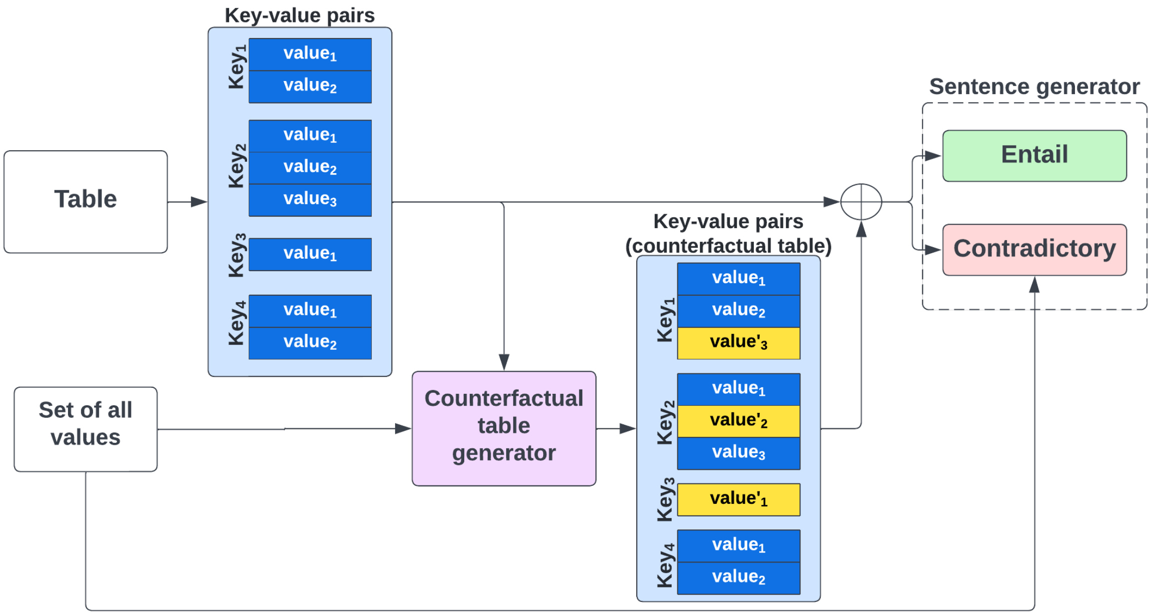

Figure 1: Framework for hypothesis generation in AUTO-TNLI. **yellow** represents values that are changed in the counterfactual tables.

Reasoning	Category	Template-Rules	Table-Constraints
Temporal	Person	<*Person*> was born in a leap year. <*Person*> died before/after <*Died:Year*>	Born Date ≤ Death Date
Numerical	Movie	<*Movie*> was a "hit **if** <*Box Office*> − <*Budget*> **else** flop" <*Movie*> had a Box Office collection of <BoxOffice>	Budget ≥ 0
Spatial	Movie	<*Movie*> was released in <*Release1:Loc*>*, "X" months before/after* <*Release2:Location*>	Release1:Location ≠ Release2:Location
KCS	City	The governing of <City> is supervised by <*Mayor*> <*Mayor*> is an important local leader of <City>	Lowest Elevation ≤ Highest Elevation

Table 2: Rules and Constraints are classified into specific areas of reasoning, as indicated in the table. A few examples of rules and constraints have been provided for each category. <Died:Year> indicates that the year value is extracted from <Died> , whereas <Release1:Location> indicates that the location is extracted from a single key-value pair in <Release>. KCS denote knowledge and common sense reasoning in this context.

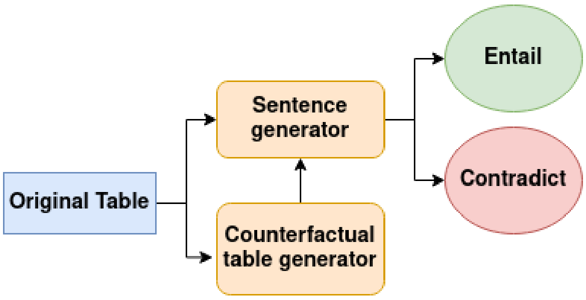

Figure 2: Hypothesis generation via original and counterfactual tables for AUTO-TNLI dataset.

ble 2 provide examples of logical rules used to create templates. We denote the category of a table as ***Category*** and the table row keys of as <*Key*>.

Frequently, these key-based reasoning rules generalize effectively across several categories. For example, the temporal reasoning rule based on the date-time type could be modified minimally to work for <*Release Date*> of category ***Movies*** tables, as well as the <*Established Date*> key of category ***University*** tables, in addition to the <*Born*> of category ***Person*** table in 2. Additionally, reasoning rules can be expanded to incorporate multi-row entities from the same table's data, as illustrated in Table 2 for the numerical reasoning type. Other examples for the same are "The elevation range of <*City*> is <*HighestElevation*> − <*LowestElevation*>" for category ***City*** table, "<*SportName*> was held at <*location*> on <*date*>" for ***Sports*** category table.

2.2 Rational Counterfactual Table Creation

One can create counterfactual tables for a particular category with total possibles n keys by modifying an existing category table with k keys ($n >= k$) as follows: (a.) keep the row as it without any change, (b.) adding new value to an existing key, (c.) substituting the existing key-value with counterfactual data, (d.) deleting a particular key-value pair from the table, (e.) and add a missing new keys (i.e. a key from $(n - k)$), (f.) and adding a missing key row to the table.

For creating counterfactual tables, for each row of existing, a subset of operation is selected at a random each with a pre-decided probability p (a hyper-parameters). While creating these tables, we imposed a essential key-specific constraints to ensure logical rational in the generated sentences. E.g. in the example Table 1, for the counterfactual table of *Janet Leigh (Counterfactual)*, the <*Born*> is kept similar to original of *Janet Leigh (Original)*, whereas <*Died*> has been substituted for another *Person* table, while ensuring the constraint BORN DATE < DEATH DATE i.e. Jan 13, 1994 (Died Date of Counterfactual Table) is after July 6, 1927 (Born Date of Counterfactual Table)). Without the following the constraint that BORN DATE < DEATH DATE, the table with become rationally incorrect or self contradictory.

2.3 Paraphrasing of Premise Tables

For each key for of a given category, we create at least three simple paraphrased sentences of the key-specific template. E.g. for <*Alma Mater*> from ***Person***, possible paraphrases can be "<*PersonName*> earned his degree from <*AlmaMater*>", "<*PersonName*> is a graduate of <*AlmaMater*>", and "<*PersonName*> is a almamater of <*AlmaMater*>".

2.4 Automatic Table-Hypothesis Generation

Once the templates are constructed, they can be used to automatically fill in the blanks from the entries of tables and create logically rational hypothesis sentences. On each turn, the machine creates unique counter-factual tables because the values to be added/substituted/deleted are randomly selected from a universal collection of same row-value types distinct values, i.e. constructed using all values entries for that key in all training tables of the dataset.

Contradict sentences are constructed similarly to entail sentences by picking a random item from the universal set and substituting it for the original value while adhering to the key-specific constraints. We ensure that similar template with minimal token alteration is used to create entail contradict pair. This way of creating entail and contradiction statement with lexically overlapping tokens ensure that, future model trained on such data won't adhere spurious correlation from the tabular NLI data i.e. minimising the hypothesis bias problem (Poliak et al., 2018).

3 The Auto-TNLI Dataset

We extend the InfoTabs to construct Auto-TNLI using the framework described in Section 2. InfoTabs (Gupta et al., 2020) consists of a pair of sentences: a hypothesis statement grounded and inferred on premise table take extracted form Wikipedia Infobox table across multiple diverse categories. We construct the Auto-TNLI dataset from a subset of the InfoTabs dataset (11 out of 13 total categories), which includes the original table plus five counterfactual tables corresponding to each original table, for a total of $10,182$ tables. We retrieve 134 keys and 660 templates in total, which we utilize to generate $1,478,662$ sentences. However, unlike InfoTabs, which contains three labels, Entail, Contradict and Neutral, Auto-TNLI contains only two labels Entail and Contradict.

Statistic Metric	Numbers
Number of Unique Keys	134
Average number of keys per table	12.63
Average number of sentences per table	164.51

Table 3: Auto-TNLI Statistics.

As previously reported in the original InfoTabs paper by Gupta et al. (2020), annotators are biased toward certain keys over others. For example, for the category ***Company***, annotators would create more sentences for the key <*Founded by*> than for the key <*Website*>, resulting in an inherent hypothesis bias in the dataset. While creating the templates for Auto-TNLI, we ensure that each key has a minimum of two hypotheses and a minimum of three (> 3) premise paraphrases, which helps in mitigating hypothesis bias. To address the labeling issue of inference class imbalance, we construct approximately 1:1 Entail to Contradict hypothesis.

We observe that the majority of additional human labor required to build such sentences is spent on the set of key-specific rules and constraints that ensure the sentences are grammatically accurate and the counter-factual tabular data is logically consistent, i.e. not self-contradictory. Table 3 details the number of unique keys, the minimum/maximum/average number of keys, and the total number of sentences per table in Auto-TNLI. As can be observed, the system generates a large amount of Auto-TNLI data in comparison to limited InfoTabs while using only few human constructed templates with key-specific rules and constraints. Table 4 shows a qualitative comparison between Auto-TNLI and InfoTabs.

	Features		
Dataset	**Coverage**	**Scalability**	**Diversity**
InfoTabs	✗	✗	✓
Auto-TNLI	✓	✓	✓

Table 4: Comparison between Auto-TNLI & InfoTabs. Coverage - implies how well the keys are covered. Scalability - implies whether the amount of data can be increased. Diversity - implies diversity across category.

We have chosen InfoTabs as it has three evaluation sets α_1, α_2 and α_3 in addition to the usual training and development sets. The α_1 set is lexically and topic-wise similar to the train set, in α_2 the hypothesis are lexically adversarial to the train set and in α_3 the tables are from topics not in the train set. Moreover it has several reasoning types such as multi-row reasoning, entity type, negation, knowledge & common sense etc. InfoTabs has all three labels Entail, Neutral and Contradict compared to just two labels in other datasets such as TABFACT.

4 Experiment and Analysis

In this section, we detail the several experiments that were conducted, as well as their outcomes and analyses.

We use RoBERTa$_{\text{BASE}}$[1] (Liu et al., 2019) (12-layer, 768-hidden, 12-heads, 125M parameters) as our model for all of our experiments. Neeraja et al. (2021) shows data augmentation techniques

[1] Experiments on the development set showed that RoBERTa$_{\text{BASE}}$ outperforms other pre-trained language models. BERT$_{\text{BASE}}$ and ALBERT$_{\text{BASE}}$ reached an accuracy of 63% and 70.4% respectively.

Setting	Train-Data	Cat-Rand	Cross-Cat	Key	No-Para	Cross-Para	Entity
InfoTabS	Random	50.00	50.00	50.00	50.00	50.00	50.00
	w/o finetuning	50.00	49.64	50.17	49.77	49.75	49.78
	InfoTabS	66.17	63.86	**65.41**	65.15	65.12	63.66
	MNLI	67.15	64.95	64.79	65.33	65.33	62.2
	MNLI +InfoTabS	**69.28**	**65.9**	65.25	**66.41**	**66.39**	**65.02**
MNLI +InfoTabS	Hypothesis-Only	53.74	55.1	58.82	66.47	66.86	56.36
	Auto-TNLI	78.74	77.94	82.39	90.06	89.38	74.94
	MNLI +Auto-TNLI	83.82	78.95	84.71	**91.17**	**90.57**	**77.66**
	MNLI +InfoTabS +Auto-TNLI	**83.62**	**80.78**	**85.23**	90.98	90.03	77.19

Table 5: Performance (accuracy) on Auto-TNLI with RoBERTa$_{\text{BASE}}$ model across several evaluation splits with / without fine-tuning on Auto-TNLI. **bold** - represents max across rows i.e. best train/augmentation setting.

that uses MNLI data for pre-training acts as implicit knowledge and enhances the model performance for InfoTabS. Therefore, we also explore implicit knowledge addition via data augmentation. In particular, we explored the following models: (a) RoBERTa$_{\text{BASE}}$ fine-tuned using the Auto-TNLI dataset (b) RoBERTa$_{\text{BASE}}$, fine-tuned on the MNLI dataset and the Auto-TNLI dataset (MNLI + Auto-TNLI). Additionally, we also explore performance with RoBERTa$_{\text{BASE}}$ model fine-tuned sequential on all three MNLI, Auto-TNLI and InfoTabS dataset.

Overall, we address the following two research questions through our experiments:

RQ1: (a) How challenging is Auto-TNLI for the task of tabular NLI (TNLI)? (b) Is fine-tuning with the training set beneficial for the TNLI task?

RQ2: (a) Can Auto-TNLI be effectively utilised for data augmentation, i.e. implicit knowledge addition, to enhance performance over InfoTabS test-sets? (b) How useful is Auto-TNLI in scenarios of limited supervision?

4.1 Using Auto-TNLI as TNLI dataset

In this section, we access how challenging our Auto-TNLI is in comparison to the InfoTabS datasets (i.e. RQ1).

Data Splits: We first construct several train-dev-test splits of Auto-TNLI such that: (a) splits have table from different domains (categories)[2] (b) splits have unique table row-keys, (c) premises in splits are lexically diverse. For the category-wise splits, we explore two ways (a) we divided categories randomly into train-dev-test. (b) we construct the splits after doing a cross-category performance analysis (refer §7 in the Appendix). In cross-category analysis we get all premise-hypothesis pairs generated from tables in one category (for example : *person*) and train our model on just this data. After this we test on premise-hypothesis pairs generated from all other categories (for example : *city*, *movie* etc.) one-by-one. We keep the categories that are difficult for the model to solve in the test set, this is achieved by calculating the number of times a category's accuracy falls below a certain threshold [3] and then selecting the top few categories. We kept *book*, *paint*, *sports & events*, *food & drinks*, *album* in train-set, *person*, *movie*, *city* in dev-set and *organization*,*festival*, *university* in test-set.

For key-wise split, we explore two approaches (a) we divide the keys randomly into train-dev-test. (b) we decided splits based on the associated keys-values named entities type namely - *person*, *person type*, *skill*, *organization*, *quantity*, *date time*, *location*, *event*, *url*, *product* after cross-entity performance analysis.. Similar to cross-category analysis above, here we get all premise-hypothesis pairs corresponding to keys in a single entity, for example let's say we choose the entity *person* and it includes the keys *written by*, *mayor*, *president* etc. then we get all premise-hypothesis pairs corresponding to these keys and train on them. After this we test on premise-hypothesis pairs corresponding to all other entities (for example : *persontype*, *skill*) one-by-one. We keep the entities that are difficult for the model to solve in the test set, this is achieved by calculating the number of times a entity's accuracy falls below a certain threshold [4] and then selecting the top few entities. We kept the *url*, *event*, *person type*, *skill*, *product* in train-set, *quantity*, *other*, *person* in dev-set and *date time*,*organization*, *location* in test-set.

Finally, for the lexical diversity, we splits via paraphrasing premise. Here too, we explore two different strategies (a) premises in train, dev and

[2] by table domain/categories we refer to table entity types e.g. "Person", "Album", and others

[3] we choose the threshold as 80% [4] we choose the threshold as 80%

test are not paraphrased i.e. have similar templates. (b) premises in train, dev and test are lexically paraphrased i.e. have distinct templates.

Using AUTO-TNLI **only for Evaluation (RQ1a):** We first explore how challenging is AUTO-TNLI is used as an evaluation benchmark dataset. To explore this. we compare the performance of pre-trained RoBERTa$_{\text{BASE}}$ model in four distinct settings, as follows (a.) without (w/o) fine-tuning, (b.) fine-tuned with INFOTABS, (c.) fine-tuned with MNLI, (d.) fine-tuned over both MNLI and INFOTABS in order and and evaluate it on AUTO-TNLI test-sets splits. For finetuning on MNLI and INFOTABS dataset, we only consider the ENTAIL and CONTRADICT while excluding the NEUTRAL label instances for training purposes.

Analysis. Table 5 shows a comparison of accuracy across all augmentation settings. The best is obtained when using both MNLI and INFOTABS for training. In the cases where we have used some fine-tuning with MNLI or INFOTABS we observed an average accuracy of 67.5% comparing this with zero-shot accuracy for INFOTABS where we observed an accuracy of 58.9% we can see that semi-automatically generated data is still pretty challenging to solve.

Using AUTO-TNLI **for both Training and Evaluation (RQ1b):** Next, we explore if providing supervision improves the performance on the AUTO-TNLI evaluation sets. To explore this, we compare the performance of pre-trained RoBERTa$_{\text{BASE}}$ model in two distinct settings, where we fine-tune on train set (a.) of AUTO-TNLI, (b.) of both MNLI and AUTO-TNLI in order and evaluate on AUTO-TNLI test-sets. Here too, we exclude the NEUTRAL label instances from MNLI.

Analysis. Table 5 shows a comparison of performance (accuracy) across all augmentation settings. For all splits, except paraphrasing RoBERTa$_{\text{BASE}}$ is 80% accurate on average, which shows that our semi-automated dataset AUTO-TNLI is as challenging as INFOTABS (Gupta et al., 2020), which has an average accuracy of 70% across all splits and is manually human-generated and is one-tenth the size of AUTO-TNLI. Pre-finetuning with MNLI as augmented data (i.e. implicit knowledge) only improves the performance by 2%.

4.2 Using AUTO-TNLI for Data Augmentation

We explore if AUTO-TNLI can be used as an augmentation dataset for INFOTABS (i.e. RQ2). Since INFOTABS include all three ENTAIL, NEUTRAL and CONTRADICT labels, whereas AUTO-TNLI include only ENTAIL and CONTRADICT labels, we explore the inference task as a two-stage classification problem. In first stage, we train a RoBERTa$_{\text{BASE}}$ classification model to predicts whether a hypothesis is NEUTRAL vs NON-NEUTRAL (either ENTAIL or CONTRADICT). In second stage, we fine-tune a separate RoBERTa$_{\text{BASE}}$ model to further classify the NON-NEUTRAL prediction instances from stage one into ENTAIL or CONTRADICT label. Figure 3 illustrate the two-stage classification approach.

Comparison Models. For first-stage we consider two training strategies: (a.) only train on INFOTABS, (b.) pre-finetune on both MNLI followed by training on INFOTABS. We consider multiple data augmentation technique for second stage training where we augment (a.) **Orig:** the AUTO-TNLI without counterfactual table instances, (b.) **Orig+Counter:** AUTO-TNLI including counterfactual table instances[5], (c.) **MNLI + Orig:** both MNLI and AUTO-TNLI without counterfactual table instances (i.e. (b.)), (d.) **MNLI + Orig + Counter:** both MNLI and AUTO-TNLI including counterfactual table instances (i.e. (c.)). . Additionally, we compare all above methods with (e.) **No Augmentation** i.e. the approach where we does not augment any additional data.

Evaluation Set. For evaluation, we utilize the INFOTABS test sets which include all three inference labels. In addition to standard development and a test split (α_1), INFOTABS also include two adversarial test splits namely α_2 and α_3. Adversarial test splits α_2 contain instances that are lexically similar to α_1, except that the ENTAIL and CONTRADICT labels (and vice-versa) are manually flipped by human annotator via minimal perturbation in hypothesis sentences. E.g. in the example table 1 if hypothesis sentence *Janet Leigh was born* before *1940* is ENTAIL, then in α_2 after perturbation the instance became *Janet Leigh was born* after *1940* with label as CONTRADICT. The test set α_3 is a zero-shot evaluation set, which consists of premise tables from different domain with minimal key overlaps with the training set premise tables. To better handle, α_2 and α_3 test-sets, we include counterfactual table and hypothesis in AUTO-TNLI.

[5] We take five counterfactual table for each original table

		E vs C				
N vs NN	Test-split	No Augmentation	Orig	Orig+Counter	MNLI+Orig	MNLI+Orig+Counter
InfoTabS	dev	71.06	70.72	71.39	**72.28**	72.22
	α_1	67.72	67.56	69.33	68.78	**69.89**
	α_2	59.11	59.22	58.94	59.5	**61.28**
	α_3	56.94	56.94	58.17	58.33	**58.61**
MNLI +InfoTabS	dev	70.67	70.89	71.44	72.56	**72.67**
	α_1	68.94	68.83	70.56	70.67	**72.00**
	α_2	60.56	60.83	60.5	61.11	**62.50**
	α_3	58.44	57.72	59.11	**60.06**	59.94

Table 6: Performance (accuracy) of combine stage one RoBERTa$_{\text{BASE}}$ (i.e. Neutral vs Non-Neutral) and stage two RoBERTa$_{\text{BASE}}$ (i.e. Entail vs Contradict) classifier across several data augmentation settings. Here, for stage one we explore also explore pre-fine tuning on MNLI data. **bold** - represents max across columns i.e. the best augmentation setting.

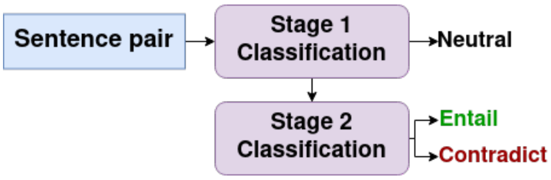

Figure 3: Two stage classification approach.

Supervision Scenarios We analyse the effect of using Auto-TNLI as augmentation data for InfoTabS in two setting (a) **Complete Supervision** where we use complete InfoTabS training set for final fine-tuning (b) **Limited Supervision** where we use limited InfoTabS supervision for second stages. We explore using 0% (i.e. no fine-tune), 5%, 10%, 15%, 20% and 25% of InfoTabS training set for final fine-tuning. .

1. Complete InfoTabS **Supervision (RQ2a)** Table 6 shows a comparison of accuracy across all augmentation settings.

In the **first case**, when the first stage is only trained on InfoTabS, we observe an improvement of 1.6% and 1.2% percentage in α_1 and α_3 test-set through direct Auto-TNLI data augmentation base pre-finetuning (Orig+Counter) in comparison with no augmentation i.e. direct InfoTabS fine-tuning. We didn't see any substantial improvement in α_2 performance. Fine-tuning with MNLI followed by Auto-TNLI (with counterfactual tables) further improve the performance by 0.6%, 2.0%, and 0.45% on α_1, α_2 and α_3 respectively.

For **second case**, when the first stage is trained on both MNLI, followed by InfoTabS, we observe an improvement of 1.60% and 0.67% percentage in α_1 and α_3 test-set through direct Auto-TNLI data augmentation base pre-finetuning (Orig+Counter) in comparison with no augmentation i.e. direct InfoTabS fine-tuning. Here too, we didn't see any substantial improvement in α_2 performance. Finetuning with MNLI followed by Auto-TNLI (with counterfactual tables) further improve the performance by 1.44%, 1.94%, and 0.83% on α_1, α_2 and α_3 respectively.

Ablation Analysis - Independent Stage-1 and Stage-2 Performance: We also did an ablation study to access the performance of individual RoBERTa$_{\text{BASE}}$ models of both stages. Table 8, show the performance for stage one classifier i.e. Neutral vs Non-Neutral. We observe that by adding MNLI data for augmentation substantially improve the performance by 1.89%, 2.28%, and 2.05% for α_1, α_2 and α_3 respectively.

Table 7 shows the comparison between all settings of stage-2. In stage-2 adding counterfactual tables improve the performance by 2.75% and 1.42% in α_2 and α_3 respectively. We didn't see any substantial improvement in α_2 performance. If we pre-finetune further with MNLI along with Auto-TNLI we further get an improvement of 5.42%, 3.33% and 2% in α_1, α_2, and α_3 respectively.

2. Limited InfoTabS **Supervision (RQ2b)** In this setting, we analyse the effect of limiting InfoTabS supervision for the second stage i.e. Entail vs Contradict. We explore using 0% (i.e. no fine-tune), 5%, 10%, 15%, 20% and 25% of InfoTabS training set for fine-tuning. Table 9 shows the performance for every augmentation settings. The table report average result over three random samples from Auto-TNLI. We observe that augmenting with Auto-TNLI improve performance for all percentages. Furthermore, the improvement is much more substantial for lower than higher percentages. Here too, the best performance are obtained via fine-tuning with MNLI followed by Auto-TNLI for all percentages.

Test-split	No-Augmentation	Orig	Orig+Counter	MNLI+Orig	MNLI+Orig+Counter
dev	77.5	77.83	78.08	**80.75**	80.25
α_1	73.58	73.83	76.33	76.5	**79.00**
α_2	56.92	57.42	56.92	58.42	**60.25**
α_3	70.58	69.42	72	**73.08**	72.58

Table 7: Performance (accuracy) of stage two RoBERTa$_{\text{BASE}}$ (i.e. Entail vs Contradict) classifier across several data augmentation settings. **bold** same as Table 6.

Test-split	No-Augmentation	MNLI
dev	84.11	**84.50**
α_1	82.94	**84.83**
α_2	85.33	**87.61**
α_3	73.17	**75.22**

Table 8: Performance (accuracy) of stage one RoBERTa$_{\text{BASE}}$ (i.e. Neutral vs Non-Neutral) across several data augmentation settings. Here, No-Augmentation means InfoTabS, and MNLI means MNLI + InfoTabS. **bold** same as Table 6.

5 Discussion

Why Semi-Automatic Framework? By examining the two diametrically opposed of frameworks, namely a Human and an Automatic Annotation Framework, we may see a plethora of issues with both. Manually created data is prohibitively expensive and demands a great deal of human effort, limiting the ability to create large-scale databases Additionally, humans have a propensity to create artificial patterns when manually creating a dataset, such as not giving all keys same importance (explained in Section 3). At the same time, while autonomous data generation is computationally efficient, it has a number of limitations, including the inability to generate linguistically difficult sentences and the difficulty of incorporating reasoning into dataset. Because most deep learning models perform better with more data, it is critical to produce large-scale datasets at a reasonable cost while retaining data quality. With this in mind, we presented a "semi-automatic" architecture with the following benefits: 1. It simplifies the creation of large-scale datasets. Using only 660 templates, we can generate 1,478,662 premise-hypothesis pairings from around 10,182 tables. 2. The framework may be reused with additional tabular data as long as the structure is preserved. 3. It enables the creation of linguistically and lexically diverse datasets. 4. As shown in Section 3, hypothesis bias can be minimized by establishing an adequate number of diverse templates for all key of each category. 5. The premises have been paraphrased in three different ways to bring the necessary lexical diversity.

Why Counterfactual Table Generation? Tabular dataset is inherently semi-structured in nature. Therefore, for each category table having a specific set of keys. This enables us to create key-specific templates based on the entity-types of keys (Neeraja et al., 2021), which could be applied to million of tables of given category. Furthermore, as explained in section 3, the templates also generalize across keys with similar value type across categories. All this is only possible due to semi-structure nature of tabular data. Using counterfactual tables provides the model with more linguistically comparable but oppositely labeled data to learn from, guaranteeing that the model can learns beyond the superficial textual artifacts and so becomes more resilient. As a result, when counterfactual data is included in the Auto-TNLI, we observe performance improvement throughout all experimental settings. This learning is further supported by the findings for better gains in α_2, which comprises instances of linguistically comparable but oppositely labeled data instances.

6 Related Work

Tabular Reasoning. There has been considerable work on solving NLP tasks on semi-structured tabular data, such as tabular NLI (Gupta et al., 2020; Chen et al., 2020b), question-answering task (Pasupat and Liang, 2015; Krishnamurthy et al., 2017; Abbas et al., 2016; Sun et al., 2016; Chen et al., 2021a, 2020c; Lin et al., 2020; Zayats et al., 2021; Oğuz et al., 2020, and others) and table-to-text generation (Parikh et al., 2020; Nan et al., 2021; Yoran et al., 2021; Chen et al., 2021b).

Similar to our data setting, some recent papers have also proposed ideas for representing Wikipedia relational tables, some such papers are TAPAS (Herzig et al., 2020), TaBERT (Yin et al., 2020b), TABBIE (Iida et al., 2021),TabStruc (Zhang et al., 2020), TabGCN (Pramanick and Bhattacharya, 2021) and RCI (Glass et al., 2021). Some papers such as (Yu et al., 2018, 2021; Eisenschlos et al., 2020; Neeraja et al., 2021; Müller

Test-split	Train (%)	No Augmentation	Orig	Orig+Counter	MNLI+Orig	MNLI+Orig+Counter
dev	0	50.25	59.58	52.58	**62.67**	60.75
	5	65.31	69.92	69.86	70.81	**71.11**
	10	67.53	72.08	69.83	**74.83**	73.42
	15	69.47	71.69	73.61	**75.28**	74.42
	20	71.28	73.61	72.47	**74.11**	**74.11**
	25	70.21	72.88	74.54	**74.71**	74.63
α_1	0	49.92	59.42	52.42	61.58	**62.33**
	5	65.75	69.08	68.89	70.72	**70.92**
	10	67.58	71.42	69	72.58	**74**
	15	69.14	70.69	70.83	73.28	**74.25**
	20	71.53	72.47	72.39	74.03	**74.61**
	25	69.75	72.38	73.75	74.5	**75.13**
α_2	0	50.17	59.00	59.75	61.17	**61.67**
	5	43.81	54.92	53.53	56.25	**58.03**
	10	47.92	54.08	54.5	**58.83**	56.75
	15	47.31	54	53.03	56.89	**57.42**
	20	49.17	54.03	54.44	**56.89**	55.75
	25	49.79	56.33	55.25	**59**	58.42
α_3	0	49.42	59.25	56.33	64.67	**63.92**
	5	57.72	63.47	63.5	68.06	**68.14**
	10	60.67	65.75	62.5	**71.58**	67.67
	15	64.42	65.69	68.47	70.03	**71.11**
	20	65.22	67.03	67.81	70.39	**71**
	25	64.08	67.17	67.42	70.46	**70.92**

Table 9: Performance (accuracy) of RoBERTa$_{\text{BASE}}$ (i.e. ENTAIL vs CONTRADICT) classifier with various data augmentation for limited supervision setting i.e. varying percentage of INFOTABS training data. The average standard deviation across 3 runs is 1.36 with range varying from 0.5% to 3.5%. **bold** same as Table 6.

et al., 2021, and others) study the improvement of tabular inference by pre-training.

Tabular Datasets. Synthetic creation of dataset has long been explored (Rozen et al., 2019; Müller et al., 2021; Kaushik et al., 2020; Xiong et al., 2020, and others). For tabular NLI in particular, the datasets can be categorized into 1) Manually created datasets (Gupta et al., 2020) with manually creates both hypothesis and premise, (Chen et al., 2020b) manually creates the hypothesis while premise is automatically generated 2) Synthetically created semi-automatically generated datasets which completely automate data generation (Müller et al., 2021; Chen et al., 2020a,d,a). Several works such as Poliak et al. (2018); Niven and Kao (2019); Gururangan et al. (2018); Glockner et al. (2018); Naik et al. (2018); Wallace et al. (2019) have shown that models exploit spurious patterns in data. Similar to Nie et al. (2019); Zellers et al. (2018); Gupta et al. (2020) authors investigate impacts of artifacts in dataset by creating adversarial testsets.

INFOTABS (Gupta et al., 2020) pairs Wikipedia infoboxes with human-generated hypotheses (annotated via Amazon's Mechanical Turk). However, this dataset comes at a high cost of roughly $10K$ dollars for just $23k$ occurrences. Additionally, the small dataset size results in model overfitting, i.e., memorization problems. One technique to circumvent these constraints is to employ automatically produced sentences, such as TABFACT (Chen et al., 2020b) has done. Fully automated frameworks, on the other hand, have their own set of limitations: (a) The generated sentences are not complex in the manner that natural language sentences is. (b) While the rules in TABFACT are focused on simple numerical reasoning, it was demonstrated in Chen et al. (2020a) that when models were trained on the dataset LogicNLG to generate sentences with more than just superficial reasoning, only 20% of the generated sentences were deemed logically correct by humans.

7 Conclusion

This paper introduced a semi-automatic framework for generating data from tabular data. We looked into using this dataset as standalone data and as an augmentation dataset with minimal fine-tuning data. We also show how to turn this dataset into a 3-label dataset and utilize it as an augmentation dataset for INFOTABS using a two-stage classification algorithm. Finally, we've shown that this dataset can be beneficial in adversarial circumstances. Future work in this direction includes creating further lexically diverse and robust datasets and exploring whether adding neutrals can improve this data.

Acknowledgement

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

References

Faheem Abbas, Muhammad Malik, Muhammad Rashid, and Rizwan Zafar. 2016. Wikiqa — a question answering system on wikipedia using freebase, dbpedia and infobox. pages 185–193.

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021a. KACE: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527, Online. Association for Computational Linguistics.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021b. Open question answering over tables and text. In *International Conference on Learning Representations*.

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

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

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

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

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

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

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

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

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

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

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

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark. Association for Computational Linguistics.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

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

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

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

Barlas Oğuz, Xilun Chen, Vladimir Karpukhin, Stanislav Peshterliev, Dmytro Okhonko, M. Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

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

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

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

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 771–782.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

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

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

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

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

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

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2895–2906, Online. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

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

A Cross-Category Analysis

We perform an analysis of how the semi-automatic data created performs across categories i.e training on one category and then testing on the rest. This gave an idea of how training on data from one category generalizes over the rest. In Table 10 we have shown the accuracy when our model is trained on the categories written in rows and tested on the categories given in the columns.

Here we observed that except some categories like *Sports & Events*, *Album* and *City* the cross category accuracy is pretty high among the rest. *Album* seems to be quite a tough category with all categories giving a low cross-category accuracy when tested on it. *City* gave a challenging test set when trained on *Sport & Events*. *University* is the toughest test set for *Album*. When used as a test-set, *City* gave the least accuracy against *Sports & Events*, *Album* gives the least accuracy against *Paint*, *University* gave the least accuracy against *Sports & Events* and for the rest *Album* gave the least accuracy.

B Cross-Entity Analysis

We perform an analysis of how the semi-automatic data created performs across entities i.e training on one entity and then testing on the rest. This gave an idea of how training on data from one category generalizes over the rest. In Table 11 we have shown the accuracy when our model is trained on the entity written in rows and tested on the entities given in the columns.

Here we observed that *Date & Time* is quite a tough test-set for most entities. *Quantity* is a tough test-set for *Skill* and *URL*. For *Skill* and *Person Type* are tough test-sets for *Location* and *Quantity* respectively. When used as a test-set, *URL* gave the lowest accuracy against *Person Type*, *Quantity* gave the lowest accuracy against *URL* and for the rest the *URL* gave the least accuracy.

Category	City	Album	Person	Movie	Book	F&D	Org	Paint	Fest	S&E	Univ
City	88.64	51.85	70.34	77.29	77	68.48	75.05	70.73	75.98	66.75	77.43
Album	52.92	79.35	65.2	60.28	57.38	65.75	59.16	53.48	58.8	55.75	52.9
Person	75.57	57.57	94.58	89.72	91.02	81.99	83.86	80.52	86.01	69.58	81.25
Movie	76.49	56.97	85.41	98.26	87.01	82.11	84.65	71.29	84.79	69.34	81.01
Book	54.03	53.37	76	77.69	97.84	78.68	76.81	73.51	64.94	71.62	53.76
F&D	61.79	56.72	80.67	83.24	87.55	95.82	80.46	76.49	74.61	68.71	58.03
Org	74.73	55.89	83.67	88.26	85.08	80.64	96.36	70.72	83.85	68.84	81.22
Paint	54.24	50.45	65.71	70.39	73.41	68.3	64.52	99	59.58	61.52	54.44
Fest	73.4	52.46	82.65	87.77	81.98	78.23	80.02	72.27	88.49	64.83	77.3
S&E	51.52	53.53	69.15	73.52	85.75	72.49	70.23	76.24	61.86	95.39	52.17
Univ	76.06	51.16	78.67	85.03	76.26	76.99	78.46	68.18	79.77	69.91	91.9

Table 10: Cross-category analysis of our data. red - shows the least accuracy when trained on a category and tested on another. green - the least accuracy obtained when tested on a category and trained on the others. violet - intersection of the two cases above (**F&D**- Food & Drinks, **S&E** - Sports & Events)

Entity	Person	P&T	Skill	Org	Quantity	D&T	Location	Event	URL	Product	Other
Person	98.44	81.24	85.56	84.5	68.83	61.59	84.77	84.97	76.14	86.1	78.74
P&T	70.45	98.33	68.77	67.84	55.58	55.42	64.77	78.26	58.94	67.17	71.1
Skill	79.44	88.01	93.44	79.92	53.76	57.65	78.48	89.18	73.04	82.29	73.13
Org	92.36	87.33	86.58	95.62	63.56	58.03	87.19	87.12	84.09	86.9	81.29
Quantity	82.12	61.93	67.27	71.41	91.36	63.22	78.13	77	78.97	70.71	70.62
D&T	77.27	65.01	60.18	74.98	64.39	85.87	77.28	71.19	88.93	64.78	70.02
Location	88.32	76.32	86.3	83.18	68.89	62.31	94.43	81.57	83.69	79.98	75.75
Event	86.01	76.66	79.52	79.8	66.14	57.17	79.75	97.09	79.05	77.92	75.6
URL	61	56.27	58.42	60.88	51.61	55.02	62.68	60.56	95.25	56.07	55.09
Product	88.82	84.03	87.59	85.5	67.24	62.11	87.02	89.83	77.77	98.99	77.37
Other	83.39	84.98	80.82	78.24	62.44	58.29	76.97	86.74	69.98	82.78	93.88

Table 11: Cross-entity analysis of our data. red - shows the least accuracy when trained on a entity and tested on another. green - the least accuracy obtained when tested on an entity and trained on the others. violet - intersection of the two cases above (**P&T**- Person Type, **D&T** - Date & Time)