Realistic Data Augmentation Framework for Enhancing Tabular Reasoning

Dibyakanti Kumar[1*], **Vivek Gupta**[2*†], **Soumya Sharma**[3], **Shuo Zhang**[4]
[1]IIT Guwahati; [2]University of Utah; [3]IIT Kharagpur; [4]Bloomberg;
dibyakan@iitg.ac.in; vgupta@cs.utah.edu; soumyasharma20@gmail.com; szhang611@bloomberg.net

Abstract

Existing approaches to constructing training data for Natural Language Inference (NLI) tasks, such as for semi-structured table reasoning, are either via crowdsourcing or fully automatic methods. However, the former is expensive and time-consuming and thus limits scale, and the latter often produces naive examples that may lack complex reasoning. This paper develops a realistic semi-automated framework for data augmentation for tabular inference. Instead of manually generating a hypothesis for each table, our methodology generates hypothesis templates transferable to similar tables. In addition, our framework entails the creation of rational counterfactual tables based on human written logical constraints and premise paraphrasing. For our case study, we use the INFOTABS (Gupta et al., 2020), which is an entity-centric tabular inference dataset. We observed that our framework could generate human-like tabular inference examples, which could benefit training data augmentation, especially in the scenario with limited supervision.

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). The NLI task has been extended for tabular data where it takes tables as the premise instead of sentences, namely tabular inference task. Two popular human-curated datasets for tabular reasoning, TABFACT (Chen et al., 2020b) and INFOTABS (Gupta et al., 2020) datasets, have enhanced recent research in this area.

However, human-generated datasets are limited in scale and thus insufficient for learning with large language models (e.g., Devlin et al., 2019; Liu et al., 2019a). Since curating these datasets requires expertise, huge annotation time, and expense, they cannot be scaled. Furthermore, it has been shown that these datasets suffer from annotation bias and spurious correlation problem (e.g., Poliak et al., 2018; Gururangan et al., 2018; Geva et al., 2019). In contrast, automatically generated data lacks diversity and have naive reasoning aspects. Recently, use of large language generation model (e.g., Radford et al.; Lewis et al., 2020; Raffel et al., 2020) is also proposed for data generation (e.g., Zhao et al., 2022; Ouyang et al., 2022; Mishra et al., 2022). Despite substantial improvement, these generation approaches still lack factuality, i.e., suffer hallucination, have poor facts coverage, and also suffer from token repetition (refer to Appendix §E analysis). Recently, Chen et al. (2020a) shows that automatic tabular NLG frameworks cannot produce logical statements and provide only surface reasoning.

To address the above shortcomings, we propose a semi-automatic framework that exploits the patterns in tabular structure for hypothesis generation. Specifically, this framework generates hypothesis templates transferable to similar tables since tables with similar categories, e.g., two athlete tables in Wikipedia, will share many common attributes. In Table 1 the premise table key attributes such as "Born", "Died", "Children" will soon be shared across other tables from the "Person" category. One can generate a template for tables in the Person category, such as <Person_Name> died before/after <Died:Year>. This template could be used to generate sentences as shown in Table 1 hypothesis H1 and H1^C. Furthermore, humans can utilize cell types (e.g., Date, Boolean) for generation templates. Recently, it has been shown that training on counterfactual data enhances model robustness (Müller et al., 2021; Wang and Culotta, 2021; Rajagopal et al., 2022). Therefore, we also utilize the overlapping key pattern to create counterfactual tables. The complexity and diversity of the templates can be enforced via human annotators. Additionally, one can further enhance the diversity

*Equal Contribution †Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4411–4429

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	Stanford University		**Alma Mater**	University of California	
Occupation	None		**Occupation**	Scientist	
H1: Janet Leigh was born before 1940.		E	H1^{C}: Janet Leigh was born after 1915.		E
H2: The age of Janet Leigh is more than 70.		E	H2^{C}: The age of Janet Leigh is more than 70.		C
H3: Janet Leigh has 1 children		C	H3^{C}: Janet Leigh has more than 2 children.		C
H4: Janet Leigh graduated from Stanford University		E	H4^{C}: Janet Leigh graduated from Stanford University		C

Table 1: A example of an original and counterfactual table from the "Person" category. Here, we illustrate how multiple operations can be used to alter different keys. In addition, we have shown how the labels (E - Entail, C - Contradict) for a specific hypothesis can alter. In the “Janet Leigh” example table, the first column represents the keys (e.g. Born; Died) and the second column has the relevant values (e.g. July 6,1927; October 3, 2004 etc).

by automatic/manual paraphrasing (Dagan et al., 2013) of the template or generated sentences.

To show the effectiveness of our proposed framework, we conduct a case study with INFOTABS dataset. INFOTABS is an entity-centric dataset for tabular inference, as shown in example Table 1. We extend the INFOTABS data (25K table-hypothesis pair) by creating AUTO-TNLI, which consists of 1,478,662 table-hypothesis pairs derived from 660 human written templates based on 134 unique table keys from 10,182 tables. For experiments, we utilize AUTO-TNLI in three ways (a.) as a standalone tabular inference dataset for benchmarking, (b.) as a potential augmentation dataset to enhance tabular reasoning on INFOTABS, i.e., the human-created data (c.) as evaluation set to assess model reasoning ability. We show that AUTO-TNLI is an effective data for benchmarking and data augmentation, especially in a limited supervision setting. Thus, this semi-automatic generation methodology has the potential to provide the best of both worlds (automatic and human generation).

To summarize, we make the following contributions in this paper:

- We propose a semi-automatic framework that exploits the patterns in tabular structure for hypothesis generation.
- We apply this framework to extend the INFOTABS (Gupta et al., 2020) dataset and create a large-scale human-like synthetic data AUTO-TNLI that contains counterfactual entity-based tables.
- We conduct intensive experiments using AUTO-TNLI and demonstrate it helps benchmark and data augmentation, especially in a limited supervision setting.

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

2 Proposed Framework

Our framework includes four main components: (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 tables (e.g., *movie*), the row attributes (i.e. keys) are mostly overlapping across all tables (e.g., *Length*, *Producer*, *Director*, and others). Therefore, this consistency across table benefits in writing table category specific **key-based rules** to create logical hypothesis sentences. We create such key-based rules for the following reasoning types: (a.) Temporal Reasoning, (b.) Numerical Reasoning, (c.) Spatial Reasoning, (d.) Common Sense Reasoning. Table 3 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*>. In addition, each template is paraphrased to enhance lexical diversity.

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 minimally modified to work for <*Release Date*> of category ***Movies*** tables, as well as the <*Established Date*> of category ***University*** tables, in addition to the <*Born*> of category ***Person*** in Table 3. Additionally, reasoning rules can be expanded to incorporate multi-row entities from the same table's data, as illustrated in Table 3 for the numerical reasoning type. Other examples for the same are "The elevation range of <*City*> is <*HighestElevation*> − <*LowestEleva-*

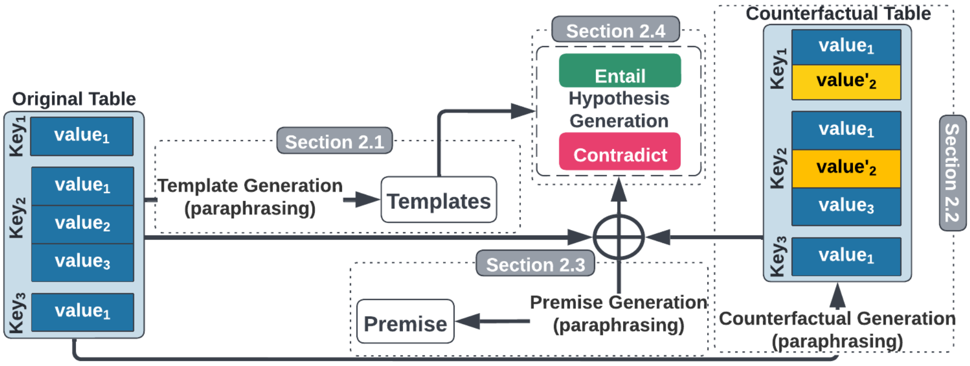

Figure 1: Our Proposed Framework. **yellow** represents modified values in the counterfactual tables.

tion>" for category ***City*** table, "<*SportName*> was held at <*location*> on <*date*>" for ***Sports*** category.

2.2 Rational Counterfactual Table Creation

We also construct counterfactual tables, as illustrated in table 1, in which the values corresponding to the original table's keys are altered. This counterfactual table contains non-factual unreal information but is consistent, i.e., the table facts are not self contradictory. Language models trained on such counterfactual instances exhibit greater robustness (Müller et al., 2021; Wang and Culotta, 2021; Rajagopal et al., 2022; Gupta et al., 2021) and prevent the model from over-fitting its pre-learned knowledge. Benefiting model in grounding and examining the premise evidence as opposed to employing spurious correlation. To create counterfactual table, we modify an original table with k keys. For a given category, these k keys constitute a subset of the n possible unique keys ($n >= k$) for that category.

To construct a counterfactual table, we modify the original table in one or more of the following ways: (a.) keep the row as it without any change, (b.) adding new value to an existing key, (c.) substituting the existing key-value with counter-factual 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-parameter).

While creating these tables, we impose an 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 fol-

Train-Data	City	Album	Person	Movie	Book	F&D	Org	Paint	Fest	S&E	Univ
Orig	78.32	67.81	92.45	97.12	96.31	92.27	92.44	98.93	87.44	82.53	85.59
Orig +Count	61.89	**68.26**	94.45	98.67	**98.72**	97.04	96.46	99.56	**93.73**	**95.68**	93.02
MNLI +Orig	**78.6**	68.12	92.89	97.74	97.21	93.19	93.06	99.36	88.12	84.18	87.03
MNLI +Orig +Count	62.32	68.01	**94.54**	**99.01**	98.46	**97.47**	**96.8**	**99.63**	93.66	95.08	**93.56**

Table 2: Category-wise results for AUTO-TNLI (**F&D**- Food & Drinks, **S&E** - Sports & Events)

Reasoning	Category	Template-Rules	Table-Constraints
Temporal	Person	<*Person*> was born in a leap year. <*Person*> died before/after <*Died:Year*>	Born Date $\leq$ 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 $\neq$ Release2:Location
KCS	City	The governing of <*City*> is supervised by <*Mayor*> <*Mayor*> is an important local leader of <*City*>	Lowest Elevation $\leq$ Highest Elevation

Table 3: 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.

lowing the constraint that BORN DATE < DEATH DATE, the table with become rationally incorrect or self contradictory.

2.3 Paraphrasing of Premise Tables

Lack of linguistic variety is a significant concern with grammar-based data generating methods. Therefore, we employ both automated and human paraphrase of premise tables to address diversity problem. For each key for of a given category, we create at least three to five 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 "<*AlmaMater*> is a alma mater of <*PersonName*>". We observe that paraphrasing considerably increases the variability across instances.

2.4 Automatic Table-Hypothesis Generation

Once the templates are constructed as discussed in §2.1, they can be used to automatically fill in the blanks from the entries of the considered tables and create logically rational hypothesis sentences. To create contradictory sentences, we randomly select a value from a collection of key values shared by all tables to fill in the blanks. This replacement ensures that the key-specific constraints, such as the key-value type, are adhered to. Furthermore, we ensure that similar template with minimal token alteration is used to create entail contradict pair. This way of creating entail and contradiction statement pairs 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). For example, for movie "Ironman" movie with rows "Budget:$140 million" and "Box-office:$585.8 million", using the template <*Movie*> was a "hit **if** <*Box Office*> $-$ <*Budget*> **else** flop" from example Table 3, one can generate hypothesis *entail*: "The movie Ironman was a hit" and *contradict*: "The movie Ironman was a flop".

3 The AUTO-TNLI Dataset

We apply our framework as described in §2 on an entity specific tabular inference dataset INFOTABS to construct AUTO-TNLI. INFOTABS (Gupta et al., 2020) consists of pairs of NLI instances: a hypothesis statement grounded and inferred on premise table is extracted from 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, which we utilize to generate $1,478,662$ sentences. However, unlike INFOTABS, which contains 3 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 4: AUTO-TNLI Statistics.

As previously reported in the original INFOTABS paper by Gupta et al. (2020), annotators are biased towards specific 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 mitigate hypothesis bias. To address the inference class imbalance labeling issue, we construct approximately 1:1 ENTAIL to CONTRADICT the hypothesis.

We observe that most 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. The counterfactual tabular data is logically consistent, i.e., not self-contradictory. Table 4 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 compared to limited INFOTABS while using only a few human-constructed templates with key-specific rules and constraints.

We have chosen INFOTABS as it has three evaluation sets α_1, α_2, and α_3, in addition to the regular training and development sets. The α_1 set is lexically and topic-wise similar to the train set, and in α_2 the hypothesis is lexically adversarial to the train set. And in α_3 the tables are from topics not in the train set. Moreover, it has multiple 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.

Human Verification To evaluate the quality and correctness of our data, we requested one of our human annotators (expert NLP Ph.D. Grad student) to assign a label to the generated hypothesis and select a score from 1 to 5 for the grammar and complexity of the sentences. The grammar score reflects how meaningful and lexically accurate the data is, and the complexity score indicates how difficult it is to label the hypothesis correctly. This was done for about 1300 premise-hypothesis pairs from AUTO-TNLI.

Statistic Metric	Numbers
Percentage of correct labels (%)	99.4
Average Grammar score (1-5)	4.89
Average Complexity score (1-5)	3.64

Table 5: Human verification Statistics.

Analysis: As observed in Table 5, humans marked 99.5% of the data as correctly labeled and gave an average score of about 4.89 out of 5 for the grammatical accuracy of the sentences. The sentences in this data also received an average complexity score of 3.64 out of 5.

4 Experiments and Analysis

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

RQ1: (a) Taking AUTO-TNLI as an evaluation set, how challenging is the TNLI task? (b) If fine-tuning on AUTO-TNLI beneficial?

RQ2: (a) Is it beneficial to use AUTO-TNLI as data augmentation for the TNLI task? (b) If so, will it also be useful in little supervision scenario?

Experiment Settings. We use RoBERTa$_{\text{BASE}}$ (Liu et al., 2019b) (12-layer, 768-hidden, 12-heads, 125M parameters) and ALBERT$_{\text{BASE}}$ (Lan et al., 2020) (12-layer, 768-hidden, 12-heads, 12M parameters) as our model for all of our experiments [1]. Neeraja et al. (2021) shows data augmentation techniques 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. Due to limited space, we report all ALBERT [2] findings in Appendix F.

4.1 Using AUTO-TNLI as TNLI dataset

In this section, we assess how challenging our AUTO-TNLI is compared 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)[3] (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 §8 in the Appendix). In the cross-category analysis, we get all premise-hypothesis pairs generated from tables in one category (for example *person*) and train our model on 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 difficult categories for the model to solve in the test set. This is accomplished by counting the number of times an category's accuracy falls below a specific threshold[4] and then selecting the entities with the highest frequency. 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 cor-

[1] Due to the large scale of the AUTO-TNLI data, we favour BASE over LARGE models for conducting efficient experiments.

[2] 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 [3] by table domain/categories we refer to table entity types e.g. "Person", "Album", and others. [4] We choose the threshold as 80%.

Training	Augmentation Strategy	Cat-Ran	Cross-Cat	Key	NoPara	Cross-Para	Entity
W/O AUTO-TNLI	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**
W AUTO-TNLI	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 6: Accuracy 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.

responding 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 select the entities that are challenging for the model in the test set. This is accomplished by counting the number of times an entity's accuracy falls below a specific threshold[4] and then selecting the entities with the highest frequency. 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 split via paraphrasing premise. Here too, we explore two different strategies (a) premises in train, dev, and 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 6 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 accuracy of 58.9%, we can see that semi-automatically generated data is still challenging.

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 6 shows a performance (accuracy) comparison across all augmentation settings. For all splits except paraphrasing, RoBERTa$_{\text{BASE}}$ achieves an average 80% accuracy. It 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%. Identical findings were also seen with ALBERT$_{\text{BASE}}$ model, c.f. Appendix F Table 19.

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-

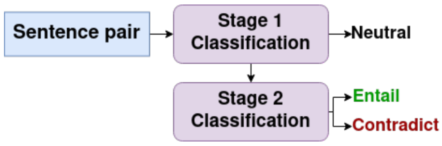

Figure 2: Two stage classification approach.

NEUTRAL prediction instances from stage one into ENTAIL or CONTRADICT label. Figure 2 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 +Count:** AUTO-TNLI including counterfactual table instances[5], (c.) **MNLI +Orig:** both MNLI and AUTO-TNLI without counterfactual table instances, (d.) **MNLI +Orig +Count:** both MNLI and AUTO-TNLI including counterfactual table instances. Additionally, we compare all above methods with (e.) **No Aug** i.e. the approach where we do not augment any additional data.

Evaluation Set. We utilize the INFOTABS test sets, which include all three inference labels for evaluation. In addition to standard development and a test split (α_1), INFOTABS also has two adversarial test splits, namely α_2 and α_3. 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 consisting of premise tables from different domains with minimal key overlaps with the training set premise tables. To better handle α_2 and α_3 test-sets, we include a counterfactual table and hypothesis in AUTO-TNLI.

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%, 15% and 25% of INFOTABS training set for final fine-tuning.

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

Split	Stage 2: Entail vs Contradict				
	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
	Stage 1: 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**
	Stage 1: 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 7: Accuracy of combine stage I i.e. NEUTRAL vs NON-NEUTRAL and stage II i.e. ENTAIL vs CONTRADICT classifiers (RoBERTa$_{\text{BASE}}$) across several data augmentation settings. Here, for stage one we also explore pre-fine tuning on MNLI data. **bold** - represents max across columns i.e. the best augmentation setting.

1. Complete INFOTABS **Supervision (RQ2a)** Table 7 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+Count) 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+Count) 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. Identical findings were also seen with ALBERT$_{\text{BASE}}$ model, c.f. Appendix F table 17.

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 9, show the performance for stage one classifier i.e. NEUTRAL vs NON-NEUTRAL. We observe that adding MNLI data for augmentation substantially improves the performance by 1.89%, 2.28%, and 2.05% for α_1, α_2, and α_3, respectively.

Split	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
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 8: Performance (accuracy) of stage two RoBERTa$_{\text{BASE}}$ (i.e. ENTAIL vs CONTRADICT) classifier across several data augmentation settings. **bold** same as Table 7.

Table 8 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. Identical findings were also seen with ALBERT$_{\text{BASE}}$ model, c.f. Appendix F Table 16 and 18.

Test-split	No Aug	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 9: 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 7.

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%, 15% and 25% of INFOTABS training set for fine-tuning. Table 10 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. In the Appendix Table 14 and 15, we present the combined stage performance on limited supervision both w and w/o MNLI pre-training. Refer the first stage results with limited

Tr(%)	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
	Development set				
0	50.25	59.58	52.58	**62.67**	60.75
5	65.31	69.92	69.86	70.81	**71.11**
15	69.47	71.69	73.61	**75.28**	74.42
25	70.21	72.88	74.54	**74.71**	74.63
	α_1 **set**				
0	49.92	59.42	52.42	61.58	**62.33**
5	65.75	69.08	68.89	70.72	**70.92**
15	69.14	70.69	70.83	73.28	**74.25**
25	69.75	72.38	73.75	74.5	**75.13**
	α_2 **set**				
0	50.17	59.00	59.75	61.17	**61.67**
5	43.81	54.92	53.53	56.25	**58.03**
15	47.31	54	53.03	56.89	**57.42**
25	49.79	56.33	55.25	**59**	58.42
	α_3 **set**				
0	49.42	59.25	56.33	64.67	**63.92**
5	57.72	63.47	63.5	68.06	**68.14**
15	64.42	65.69	68.47	70.03	**71.11**
25	64.08	67.17	67.42	70.46	**70.92**

Table 10: Performance (accuracy) of RoBERTa$_{\text{BASE}}$ (i.e. ENTAIL vs CONTRADICT i.e. second stage) 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 0.5% to 3.5%. **bold** same as Table 7.

supervision in Appendix Table 13. Appendix Figure 4 show consistency analysis.

5 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; Gupta et al., 2022), question-answering task (Zhang and Balog, 2020; Zhu et al., 2021; Pasupat and Liang, 2015; Abbas et al., 2016; Sun et al., 2016; Chen et al., 2021a, 2020c; Lin et al., 2020; Zayats et al., 2021; Oguz et al., 2022, and others) and table-to-text generation (Zhang et al., 2020b; Parikh et al., 2020; Nan et al., 2021; Yoran et al., 2022; 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), StrucBERT (Trabelsi et al., 2022), Table2vec (Zhang et al., 2019), TaBERT (Yin et al., 2020), TABBIE (Iida et al., 2021),TabStruc (Zhang et al., 2020a), TabGCN (Pramanick and Bhattacharya, 2021), RCI (Glass et al., 2021), TURL (Deng et al., 2022) and TableFormer (Yang et al., 2022). Some papers such as (Yu et al., 2018, 2021; Eisenschlos et al., 2020; Neeraja et al., 2021; Müller et al., 2021; Somepalli 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 requires manual designing table-dependent context-free grammar (CFG) (Eisenschlos et al., 2020), or require logical forms to be annotated (Müller et al., 2021; Chen et al., 2020a,d). 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 test sets. However, semi-automatic systems requiring a CFG or logical forms contains reasoning which is often limited to certain types. Creating sentences that contain other reasonings (like lexical reasoning, knowledge, and common sense reasoning) is challenging using CFG and logical forms. Our paper requires subject matter experts to create entity specific templates for each category which leads to creating sentences with multiple reasonings as well as complex reasonings.

6 Discussion

Why Counterfactual Table Generation? Tabular dataset is inherently semi-structured. Therefore, each category table has 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 millions of tables of a given category. Furthermore, as explained in §3, the templates also generalize across keys with similar value types across categories. All this is only possible due to the semi-structured nature of tabular data. Using counterfactual tables equips the model with more linguistically comparable. But oppositely labeled data to learn from, guaranteeing that the model can learn beyond the superficial textual artifacts and so becomes more resilient as shown by (Rajagopal et al., 2022; Kaushik et al., 2020). As a result, when counterfactual data is included in the AUTO-TNLI, we observe performance improvement throughout all experimental settings. This learning is further verified by the findings for better gains in α_2, which comprises instances of linguistically comparable but oppositely labeled data instances.

Why Semi-Automatic Approach? By examining the two diametrically opposed frameworks, namely a Human and an Automatic Annotation Framework, we may see many issues with both. Manually created data is prohibitively expensive and demands much human effort, limiting the ability to develop large-scale databases. Additionally, humans have a propensity to establish artificial patterns when manually creating a dataset, such as not giving all keys the same importance (explained in §3). While autonomous data generation is computationally efficient, it has many limitations. e.g., the inability to generate linguistically complex sentences and the difficulty of incorporating reasoning into the dataset. Because most deep learning models perform better with more data, producing large-scale datasets at a reasonable cost is critical while retaining data quality. With this in mind, we presented a "semi-automatic" architecture with the following benefits: (a.) 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. (b.) The framework may be reused with additional tabular data as long as the structure is preserved. (c.) It enables the creation of linguistically and lexically diverse datasets. (d.) As shown in §3, hypothesis bias can be minimized by establishing an adequate number of diverse templates for all keys of each category. (e.) The premises have been paraphrased in three ways to bring the required lexical diversity.

7 Conclusion

We introduced a semi-automatic framework for generating data from tabular data. Using a template-based approach, we generate AUTO-TNLI. We utilized AUTO-TNLI for both TNLI evaluation and data augmentation. Our experiments demonstrate the effectiveness of AUTO-TNLI and, by implication, our framework, especially for adversarial settings. For the future work, we aim to involve the creation of additional lexically varied and robust datasets and investigate whether the addition of neutrals could improve these datasets.

8 Limitations

This work has focused on entity tables, where the tabular structure and knowledge patterns are straightforward. Nevertheless, our templates technique does not generate maybe true/maybe false statements, i.e., neutral statements, as they need enhanced common sense (e.g., subjective usage) and unmentioned entity knowledge, i.e., information beyond the premise table. It is unknown how to generate good templates automatically, such as using neural generation methods rather than leveraging expert domain knowledge. Also, how these manually curated templates work when applied with more complicated tables like nested and hierarchical tables is under-explored. Theoretically, we can generate an infinite number of premise-hypothesis pairs, but the marginal utility might hurt the notion. Additionally, the zero-shot capabilities for out-of-domain tables are limited by the presumption that tables in similar categories resemble keys.

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 thank Antara Bahursettiwar for her 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.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *SIGMOD Rec.*, 51(1):33–40.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

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

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

Vivek Gupta, Shuo Zhang, Alakananda Vempala, Yujie He, Temma Choji, and Vivek Srikumar. 2022. Right for the right reason: Evidence extraction for trustworthy tabular reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3268–3283, Dublin, Ireland. 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*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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

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. 2019a. Roberta: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

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

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

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

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 Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

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

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

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.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

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

Dheeraj Rajagopal, Siamak Shakeri, Cicero Nogueira dos Santos, Eduard Hovy, and Chung-Ching Chang. 2022. Counterfactual data augmentation improves factuality of abstractive summarization. *arXiv preprint arXiv:2205.12416*.

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.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*.

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.

Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D. Davison, and Jeff Heflin. 2022. Strubert: Structure-aware bert for table search and matching. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 442–451, New York, NY, USA. Association for Computing Machinery.

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.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14024–14031.

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*.

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

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

Ori Yoran, Alon Talmor, and Jonathan Berant. 2022. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6016–6031, Dublin, Ireland. Association for Computational Linguistics.

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. 2020a. 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.

Li Zhang, Shuo Zhang, and Krisztian Balog. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1029–1032, New York, NY, USA. Association for Computing Machinery.

Shuo Zhang and Krisztian Balog. 2020. Web table extraction, retrieval, and augmentation: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(2).

Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. 2020b. Summarizing and exploring tabular data in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1537–1540, New York, NY, USA. Association for Computing Machinery.

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

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. 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 3277–3287, Online. Association for Computational Linguistics.

A Cross-Category Analysis

We analyze how the semi-automatic data created performs across categories, i.e., training on one category and evaluating 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 categories written in rows and evaluated on the categories given in the columns.

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 11: Cross-category analysis of our data. red - shows the least accuracy when trained on a category and evaluated 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)

Analysis: Here we observed that except some categories such as *Sports & Events*, *Album* and *City* the cross category accuracy is pretty high among the rest. *Album* seems to be quite a hard category with all categories giving a low cross-category accuracy when evaluated 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 analyze how the semi-automatic data created performs across entities, i.e., training on one entity and evaluating on the rest. This gave an idea of how training on data from one category generalizes over the rest. In Table 12, we have shown the accuracy when our model is trained on the entity written in rows and evaluated on the entities given in the columns.

Analysis: 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.

C First Stage Performance with Limited Supervision

The first stage classifier is used to classify NEUTRAL vs. NON-NEUTRAL. In Table 13 we have shown the accuracy for the first stage of the 2-stage classifier in the limited supervision setting with and w/o MNLI augmentation.

D Effects of augmentation with AUTO-TNLI in limited supervision

Since AUTO-TNLI only contains ENTAIL and CONTRADICT labels, to check how pretraining with AUTO-TNLI affects the results in the limited supervision setting we had to use the 2-stage classifier where (a.) No Augmentation in first stage i.e. Table 14. (b.) Augmentation with MNLI in first stage i.e. Table 15.

Analysis: As we can see in both Table 14 and Table 15 that the best is obtained by similar models in either case, with the only difference being that augmenting the first stage with MNLI helps improve the accuracy across all cases.

E Automatic Data Generation

Using GPT-J-6B, we generate 9–11 sentences per category. In total, we generated 110 sentences for 11 categories. We then classified each sentence into one of the following five classes: (a.) Correct - Both sentence and labels are correct. (b.) Factual error - Sentence is meaningful, but the label assigned to it is wrong. (c.) Overfit error - The same sentence as seen previously is generated without any lexical changes. (d.) Hallucination error - When knowledge from outside the tables provided is used to make a sentence. (e.) Repetition error - The same sentence is generated several times.

Analysis: As observed in Figure 3, out of all the 110 automatically generated hypothesis only 32.7% were *Correct* i.e. sentences were meaningful and the labels assigned to them are correct. Among the rest, about 52% had *Factual* errors in them and around 35% were *Hallucination* errors.

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 12: 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)

Tr(%)	No Aug	MNLI	Tr(%)	No Aug	MNLI
Development set			α_2 **set**		
0	**63.11**	59.28	0	64.61	**65.33**
5	75.75	**81.42**	5	78.17	**84.03**
10	76.86	**83.08**	10	79.86	**85.53**
15	78.92	**83.03**	15	81	**85.72**
20	78.83	**82.83**	20	81.58	**85.89**
25	78.92	**83.47**	25	81.81	**85.89**
α_1 **set**			α_3 **set**		
0	**62.28**	58.5	0	**62.72**	56.06
5	76.94	**81.86**	5	69.42	**72.78**
10	77.11	**82.67**	10	69.17	**72.97**
15	79.22	**82.53**	15	70.22	**73.44**
20	78.53	**82.56**	20	67.86	**73.56**
25	78.92	**82.78**	25	68.81	**74.03**

Table 13: First stage performance (accuracy) of RoBERTa$_{\text{BASE}}$ (i.e. NEUTRAL or NON-NEUTRAL) 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 1.197 with range varying from 0% to 3.14%. **bold** same as Table 7.

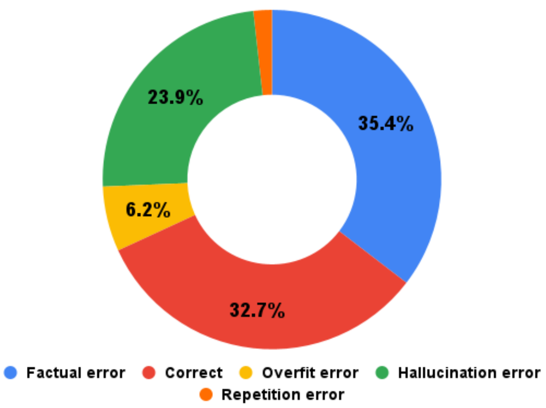

Figure 3: Percentage chart for Automatic data generation correct and error labels.

This further demonstrates that a semi-automatic approach, such as ours, is preferable, as fully automated generating techniques are not reliable.

Tr(%)	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
	Development set				
0	33.06	38.94	34.5	**39.56**	**39.56**
5	55.64	57.44	**59.11**	58.31	58.42
15	61.83	62.22	**63.44**	63.42	63.28
25	64.5	64.83	64.97	65.47	**66.11**
	α_1 **set**				
0	33	38.06	34.33	**40.28**	**40.28**
5	56.64	58.75	58.89	58.67	**59.03**
15	61.81	62.94	63.03	**64.33**	63.89
25	64.36	64.39	64.28	64.89	**65.69**
	α_2 **set**				
0	33.17	38.83	39.5	**40.5**	**40.5**
5	43.69	47.64	49.64	49.86	**51.44**
15	50.39	53.69	53.72	54.75	**54.94**
25	54.69	56	56.03	57.06	**57.94**
	α_3 **set**				
0	32.83	38.39	36.61	**41.61**	**41.61**
5	47.42	48.31	51.03	50.94	**52.31**
15	50.47	52.28	53.44	**53.86**	53.19
25	52.33	51.81	52.06	**54.25**	53.83

Table 14: Both stage performance (accuracy) of RoBERTa$_{\text{BASE}}$ (i.e. ENTAIL, CONTRADICT or NEUTRAL) classifier with various data augmentation for limited supervision setting i.e. varying percentage of INFOTABS training data w/o MNLI pretraining for first stage. The average standard deviation across 3 runs is 0.98 with range varying from 0% to 4%. **bold** same as Table 7.

F ALBERT Performance

We perform a similar analysis on ALBERT$_{\text{BASE}}$ as we have done for RoBERTa$_{\text{BASE}}$ to see if our data benefits there too. To see how robust AUTO-TNLI is when improving performance in the Augmentation setting, we perform the same experiments as RQ2a in Section 4.2. We also explore some experiments from RQ1b in Section 4.1 which are shown in Table 19.

Analysis: As we can see in Table 16 to Table 18, the trends are very similar to what we have seen in main paper Section 4.2 for full supervision setting.

Tr(%)	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
			Development set		
0	43.33	47	45.44	**47.94**	47.72
5	61.11	63.25	**64.81**	64.19	64.36
15	65.33	65.67	66.58	66.86	**66.89**
25	68.08	68.25	68.19	69.06	**69.56**
			α_1 set		
0	42.06	47.94	45.78	**48.17**	48.11
5	61.83	64.06	64.08	64	**64.36**
15	64.39	65.72	65.58	**66.97**	66.61
25	67.69	68.03	67.61	68.17	**69.03**
			α_2 set		
0	46.72	51.61	50.5	51.5	**52.06**
5	49.69	53.78	56.17	56.39	**57.67**
15	54.47	57.83	57.81	58.97	**59.14**
25	57.31	59.17	59.22	60.11	**61.08**
			α_3 set		
0	39.72	43.33	42.33	**44.72**	44.17
5	50.64	51.67	54.14	54.56	**56.22**
15	53.72	55.58	56.67	**57.19**	56.75
25	56	55.39	55.89	**58.69**	57.92

Table 15: Both stage performance (accuracy) of RoBERTa$_{\text{BASE}}$ (i.e. ENTAIL, CONTRADICT or NEUTRAL) classifier with various data augmentation for limited supervision setting i.e. varying percentage of INFOTABS training data with MNLI pretraining for first stage. The average standard deviation across 3 runs is 1.89 with range varying from 0% to 5.23%. **bold** same as Table 7.

Test-split	No-Augmentation	MNLI
dev	79.11	**85.22**
α_1	78.61	**82.83**
α_2	80.89	**85.22**
α_3	67.78	**73.94**

Table 16: Performance (accuracy) of stage one ALBERT$_{\text{BASE}}$ (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 7.

Thus our approach of semi-automatic generation is generalizable across similar models.

G Performance Across Different Reasoning Types in INFOTABS

We take the 160 pairs from development and α_3 test sets each, from INFOTABS, that have been categorised into 14 reasoning types to assess the impact of pre-training on various reasoning types, namely (a) numerical reasoning, (b) co-reference, (c) multi-row reasoning, (d) knowledge and common sense, (e) simple lookup, (f) negation, (g) lexical reasoning, (h) entity type, (i) named entities, (j) temporal reasoning, (k) subjective/out-of-table, (l) quantification, (m) syntactic alternations, and (n) ellipsis.

Split	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
			Stage 2: Entail vs Contradict		
			Stage 1: INFOTABS		
dev	60.78	61.72	62.83	**64.83**	63.89
α_1	60.89	61.33	62.78	**63.22**	63.11
α_2	49.83	53.06	51.67	55.67	**56.5**
α_3	49.39	50.11	51.72	**52.94**	51.72
			Stage 1: MNLI +INFOTABS		
dev	66.28	67.44	68.22	**70.67**	69.61
α_1	65.72	66.06	67.28	67.44	**67.5**
α_2	54	56.72	55.83	60.11	**60.89**
α_3	53.33	55.11	56.11	**57.39**	56.94

Table 17: Accuracy of combine stage I i.e. NEUTRAL vs NON-NEUTRAL and stage II i.e. ENTAIL vs CONTRADICT classifiers (ALBERT$_{\text{BASE}}$) across several data augmentation settings. Here, for stage one we also explore pre-fine tuning on MNLI data. **bold** - represents max across columns i.e. the best augmentation setting.

Split	No Aug	Orig	Orig +Count	MNLI +Orig	MNLI +Orig +Count
dev	68.92	71.25	72.33	**76**	74.5
α_1	69.42	70.92	72.92	**73.92**	73.25
α_2	47.58	52.75	50.83	58.17	**58.67**
α_3	61	64.17	66.33	**68.33**	68.08

Table 18: Accuracy of stage II i.e. ENTAIL vs CONTRADICT classifiers (ALBERT$_{\text{BASE}}$) across several data augmentation settings. **bold** same as Table 7.

The frequency of ENTAIL and CONTRADICT pairs being correctly classified is shown in Table 20 and Table 21 respectively.

Analysis: In Table 20 we observe that 9 out of 14 times in development and 12 out of 14 times in α_3-test sets MNLI + Orig + Count perform best. In Table 21 we observe that 10 out of 14 times in development set Orig + Count perform best.

H Reasoning for AUTO-TNLI

Our annotators classified all the distinct[6] templates from AUTO-TNLI into 14 reasoning types present in INFOTABS. Table 22 shows the individual reasoning type distribution across each category. The distribution statistics of reasoning types across each category is shown in Table 23. Table 24 shows that summary statistics across various reasoning types. Figure 5 gives distribution of extend of multiple reasoning in each individual examples.

Analysis: As we observe in Table 22 the cumulative frequency of reasoning types across each category is highest for Person followed by University and City and the average frequency of reasoning types across category is City followed by Person and Paint. In Table 24 we see that the cumulative

[6] templates for Provost and President are very similar so we don't consider them to be separate templates

Augmentation Strategy	Cat-Ran	Cross-Cat	Key	No-Para	Cross-Para	Entity
Random	50.00	50.00	50.00	50.00	50.00	50.00
AUTO-TNLI	77.16	69.73	81.91	86.22	**87.45**	72.75
MNLI +AUTO-TNLI	**80.28**	**76.24**	**83.1**	**88.73**	87.44	**74.53**

Table 19: Performance (accuracy) on AUTO-TNLI with ALBERT$_{\text{BASE}}$ model across several evaluation splits with fine-tuning on AUTO-TNLI. **bold** - represents max across rows i.e. best train/augmentation setting.

	Human	No Aug	Orig +Count	MNLI +Orig +Count	Human	No Aug	Orig +Count	MNLI +Orig +Count
	Development set				**α_3 set**			
numerical	11	6	**8**	**8**	14	1	3	**5**
co-reference	8	**4**	**4**	3	5	2	2	**3**
multi-row	20	**13**	11	**13**	15	6	**8**	**8**
KCS	31	18	**21**	**21**	11	6	**9**	8
temporal	19	10	15	**16**	10	6	7	**8**
syntactic-alt	0	0	0	0	2	1	1	**2**
simple-lookup	3	**3**	**3**	**3**	8	**8**	7	**8**
entity-type	6	4	**5**	4	8	3	**6**	**6**
ellipsis	0	0	0	0	1	**0**	**0**	**0**
subjective-oot	6	3	**4**	**4**	2	**1**	**1**	**1**
name-id	2	**1**	**1**	**1**	1	**1**	**1**	**1**
lexical	5	**3**	**3**	**3**	3	2	**3**	**3**
quantification	4	1	**3**	**3**	2	**2**	**2**	**2**
negation	0	0	0	0	0	0	0	0

Table 20: Frequency of labels assigned as ENTAIL in each reasoning type across 3 settings and Gold labels for INFOTABS. **bold** - represents max across rows i.e. best train/augmentation setting.

	Human	No Aug	Orig +Count	MNLI +Orig +Count	Human	No Aug	Orig +Count	MNLI +Orig +Count
	Development set				**α_3 set**			
numerical	7	**5**	**5**	**5**	14	**12**	10	7
co-reference	13	8	**10**	8	8	**6**	5	4
multi-row	17	**12**	**12**	**12**	12	**10**	8	8
KCS	24	15	**17**	16	17	**12**	**12**	**12**
temporal	25	15	**18**	15	16	**14**	11	12
syntactic-alt	0	0	0	0	0	0	0	0
simple-lookup	1	**0**	**0**	**0**	2	**2**	**2**	**2**
entity-type	6	3	**4**	**4**	9	**4**	3	1
ellipsis	0	0	0	0	0	0	0	0
subjective-oot	6	2	**3**	2	9	**5**	4	3
name-identity	1	**1**	**1**	**1**	0	0	0	0
lexical	4	**4**	3	**4**	8	**5**	4	3
quantification	6	3	**4**	**4**	4	**2**	1	**2**
negation	6	**6**	**6**	**6**	4	**3**	**3**	2

Table 21: Frequency of labels assigned as CONTRADICT in each reasoning type across 3 settings and Gold labels for INFOTABS. **bold** - represents max across rows i.e. best train/augmentation setting.

Statistics	City	Album	Person	Movie	Book	F&D	Org	Paint	Fest	S&E	Univ
numerical	19	7	28	24	16	8	19	3	14	9	5
co-reference	0	0	2	0	0	0	0	0	0	0	0
multi-row	4	0	15	5	0	7	6	7	1	1	6
KCS	21	2	45	9	3	0	24	5	0	5	27
temporal	1	6	31	5	1	0	1	4	3	6	2
syntactic-alt	23	0	6	10	2	8	6	2	14	4	28
simple-lookup	58	6	54	49	34	19	45	16	32	21	72
entity-type	0	0	54	0	0	0	1	4	0	3	1
ellipsis	0	0	20	0	0	0	1	0	0	1	0
subjective-oot	0	0	0	0	0	0	0	0	0	0	0
name-identity	0	0	6	0	0	0	3	0	0	0	0
lexical	25	0	7	20	19	3	30	2	13	6	27
quantification	13	5	19	11	11	3	8	1	10	11	3
negation	0	0	1	0	5	0	5	0	0	0	1

Table 22: Distribution of different reasoning types across all categories in AUTO-TNLI.

frequency of reasoning types across all categories is highest for simple lookup followed by lexical

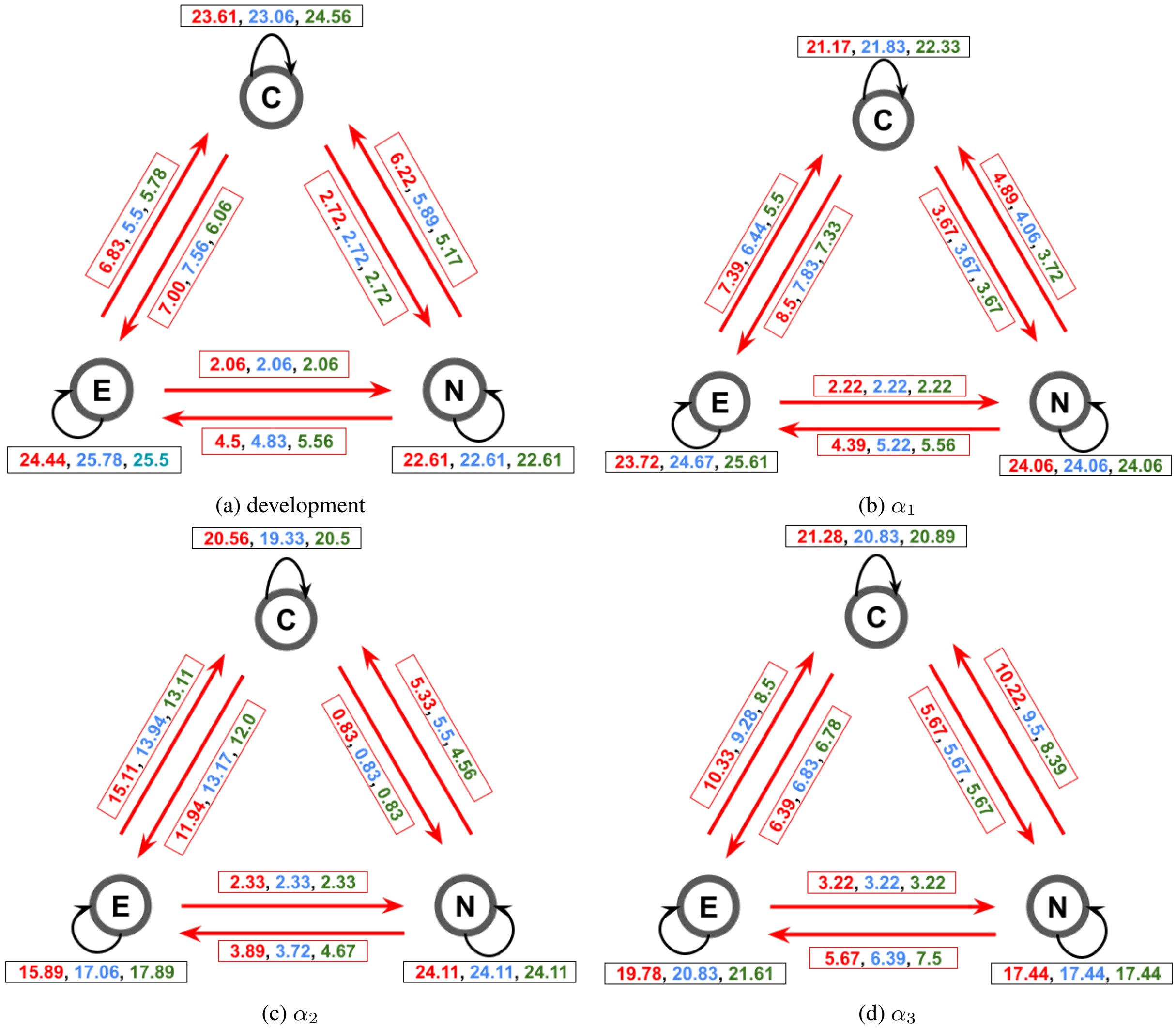

Figure 4: Consistency graphs. From left to right the values represent Red - No Augmentation, Blue - Orig+Counter, Green - MNLI+Orig+Counter.

Statistics	City	Album	Person	Movie	Book	F&D	Org	Paint	Fest	S&E	Univ
No. of reasoning	164	26	288	133	48	91	149	44	87	67	172
Avg reasoning	2.52	1.37	2.25	1.77	1.78	1.86	1.99	2.2	1.74	1.68	2.12
Max reasoning	4	2	7	3	3	4	4	4	4	3	4

Table 23: Statistics of reasoning type distribution across the different categories in AUTO-TNLI.

Reasoning	Average	Max	Min	Cumulative	Reasoning	Average	Max	Min	Cumulative
numerical	13.82	28	3	152	entity-type	5.73	54	0	63
co-reference	0.18	2	0	2	ellipsis	2	20	0	22
multi-row	4.73	15	0	52	subjective-oot	0	0	0	0
KCS	12.82	45	0	141	name-identity	0.82	6	0	9
temporal	5.45	31	0	60	lexical	13.82	30	0	152
syntactic-alt	9.36	28	0	103	quantification	8.64	19	1	95
simple-lookup	36.91	72	6	406	negation	1.091	5	0	12

Table 24: Statistics of distribution of different reasoning types across all categories in AUTO-TNLI.

and numerical which have the same frequency.

I Consistency Graphs

We perform a consistency analysis on three setting, namely No Augmentation, Orig + Count and MNLI + Orig + Count to obtain a better estimate of where pre-training with AUTO-TNLI helps improve performance in INFOTABS. In Figure 4 we have shown the consistency graphs on the 3 settings.

Analysis: We observe in Figure 4 that the model is more prone to classifying CONTRADICT as ENTAIL than the other way around in α_1 set and there

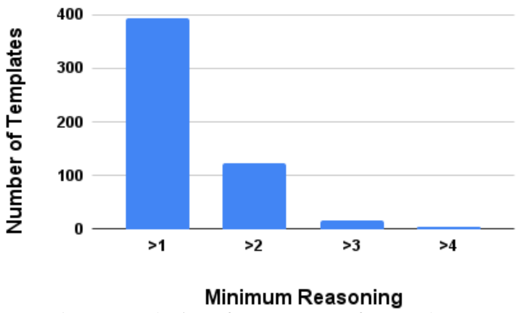

Figure 5: Cumulative frequency of templates across reasoning types in AUTO-TNLI.

is a significant improvement after pretraining with AUTO-TNLI.For α_2 and α_3 sets we can see a considerable improvement in ENTAIL being classified as CONTRADICT from pretraining on AUTO-TNLI. Pretraining on AUTO-TNLI always results in improvements overall.