TabPert: An Effective Platform for Tabular Perturbation

Nupur Jain
IIT Kanpur
nupurj@iitk.ac.in

Vivek Gupta
University of Utah
vgupta@cs.utah.edu

Anshul Rai
IIT Kanpur
anshulra@iitk.ac.in

Gaurav Kumar
IIT Kanpur
gauravkg@iitk.ac.in

Abstract

To truly grasp reasoning ability, a Natural Language Inference model should be evaluated on counterfactual data. TabPert facilitates this by assisting in the generation of such counterfactual data for assessing model tabular reasoning issues. TabPert allows a user to update a table, change its associated hypotheses, change their labels, and highlight rows that are relevant for the hypothesis classification. TabPert also captures information about the techniques used to automatically produce the table, as well as the strategies employed to generate the challenging hypotheses. These counterfactual tables and hypotheses, as well as the metadata, can then be used to explore an existing model's shortcomings methodically and quantitatively.

1 Introduction

Given factual evidence, a crucial part of NLP model reasoning capacity is evaluating whether a given hypothesis is an entailment (true), a contradiction (false), or is neutral (cannot be determined). Current transformers-based models have been shown to outperform humans on these tasks when the evidence is presented as simple unstructured text (Wang et al., 2018, 2019); however, when tested with semi-structured evidence (Gupta et al., 2020; Chen et al., 2019), such as tables, as shown in Figure 1, the very same models struggle to match human accuracy (Neeraja et al., 2021; Wang et al., 2021; Aly et al., 2021).

New York Stock Exchange	
Type	Stock exchange
Location	New York City, New York, U.S.
Founded	May 17, 1792; 226 years ago
Currency	United States dollar
No. of listings	2,400
Volume	US$20.161 trillion (2011)

H1: NYSE has fewer than 3,000 stocks listed.
H2: Over 2,500 stocks are listed in the NYSE.
H3: S&P 500 stock trading volume is over $10 trillion.

Figure 1: A tabular premise example. The table's first and second columns correspond to the keys and their associated values, respectively. The hypothesis H1 is entailed by the data in the table, H2 is a contradiction, and H3 is neutral, i.e., neither entailed nor contradictory.

Furthermore, there can be several reasons for a model's correct predictions on a particular example. For example, Poliak et al. (2018); Gururangan et al. (2018) show that multiple NLI datasets such as the SNLI and MNLI datasets (Bowman et al., 2015; Williams et al., 2018) exhibit hypothesis bias, i.e., the hypothesis-only model performs significantly better than the majority label baseline. In the context of tables, Gupta et al. (2020); Neeraja et al. (2021); Gupta et al. (2021) show that the right prediction does not always imply reasoning: there can be dataset biases in semi-structured datasets too, such as hypothesis or premise artefacts (spurious patterns) which can wrongly support a particular label.

Additionally, a model can also ignore the ground evidence and use its pre-trained knowledge for making predictions (Gupta et al., 2021). When deployed in the real world on out-of-domain (different category) or counterfactual (stories tables) examples, these models fail embarrassingly. One way to avoid this inflated performance projection is to test models on several challenging sets before deployment. For example, Gupta et al. (2020); Neeraja et al. (2021) evaluate the RoBERTa$_{Large}$ (Liu et al., 2019) models on two additional adversarial sets (hypothesis-perturbed and out-of-domain) and observe a significant performance drop. However, manually creating such challenge sets can be tricky, both in terms of the annotation cost involved and the actual annotation process, especially with tabular data of semi-structured nature.

Recently, Ribeiro et al. (2020a) have shown that one can deploy simple tricks to semi-automate

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 350–360
November 7–11, 2021. ©2021 Association for Computational Linguistics

this process of altering existing data. This semi-automated approach can then generate difficult adversarial counterfactual contrast sets, which can subsequently be utilised to perform behavioural testing of existing NLP models. However, such methods are currently only applicable to unstructured data and cannot be directly used for semi-structured text such as tables.

To fill this gap, in this work, we present TABPERT. TABPERT is an annotation platform specifically designed to work on semi-structured tabular data. For example, TABPERT can support the semi-automatic creation of tabular counterfactual data. Through TABPERT, annotators can modify tables in several ways, such as (a) *deleting information*: deleting an attribute-value pair or an existing row completely, (b) *inserting information*: inserting an attribute-value pair for an existing row or creating a fresh row, (c) *modifying information*: editing the attribute or values cells of an existing row, and (d) *modifying hypotheses or labels*: modifying an existing hypothesis and its inference label. Furthermore, each component of TABPERT can be customized to meet the individual needs of a project that necessitates tabular perturbations.

TABPERT additionally logs the strategy used to modify each attribute-value of the table. In addition to the gold label, users can manually log information about the technique used for perturbing a table-hypothesis pair and the rows relevant to the hypothesis. This information is crucial in understanding the challenges annotated data poses to the existing model, and therefore, can be utilized to probe a model's yet-unknown shortcomings systematically.

The contributions of our work can be summarised as below:

1. TABPERT can help delete, modify, and insert information in semi-structured tabular data for creating counterfactual examples.

2. TABPERT auto-logs table perturbation metadata and supports manual hypothesis modification and inference labels annotation.

3. TABPERT assists users in logging metadata, including hypothesis-related table rows and the perturbation strategy used, which is crucial for model performance analysis.

4. We present a case study for TABPERT via the generation and evaluation of a counterfactual INFOTABS dataset and RoBERTa$_{Large}$ model, respectively.

The TABPERT source code, the annotated counterfactual INFOTABS dataset, the NLI RoBERTa$_{Large}$ model, the annotation instructions with examples set, and all other associated scripts, are available at `https://tabpert.github.io`. The annotator instruction video describing TABPERT usage is accessible at `https://www.youtube.com/watch?v=sbCH_zD53Kg`.

2 Tables are Challenging

One might argue that creating a counterfactual dataset for tables is not a challenging task and that table modification can be fully automated by merely *'shuffling'* or *'inserting'* attribute values of one table row into another table row (with the same attribute) as long as they are from similar categories, e.g. shuffle *'Producer'* of one movie with *'Producer'* of another movie). One can extend this further by shuffling rows with different attributes in the same as well as different tables (same category) as long as the name-entity type for values is similar, e.g. shuffle *'Producer'* with the *'Director'* of the same or a different movie with each other.

This method, however, does not automate the updation of associated hypotheses and inference labels. Furthermore, such automated shuffling quite often flagrantly violates common-sense logical constraints. For example, a person's *'Birth Date'* must be before their *'Died Date'*, a person's *'Marriage Date'* should be after their *'Birth Date'* and before their *'Died Date'*, an album's *'Released Date'* should be after its *'Recording date'* and so on. The updated table may be self-contradictory if these constraints are not enforced. While some of these constraints can be automatically met and therefore not violated, the vast majority of them inevitably sneak through due to their enormous diversity and variance[1]. Furthermore, due to the domain-specific nature of these constraints, enforcing them automatically during perturbation is a challenging task. Keeping this in mind, automated perturbations like these are only appropriate for table initialization. Human annotators can then manually analyze and modify the initialized tables for self-consistency, i.e., no logical common sense constraint violation.

[1] In real data, these constraints are naturally satisfied.

3 TabPert Functions, Aspects, and Usability

TabPert is currently supported on common web browsers such as Google Chrome and can be installed to run locally[2]. We start with a dataset of tables along with already annotated labelled hypotheses. We utilize the InfoTabS dataset for the case study provided in this research. InfoTabS is a semi-structured natural language inference dataset that consists of entity tables and human-written hypotheses. We create three counterfactual tables (labelled A, B, and C) for each original table in the dataset. There are three main steps required for successful annotation, as described below.

3.1 Automatic Initialization

First, we initialise TabPert with original tables and counterfactual tables generated via automatic random 'shuffling' of table rows or attribute values[3]. Automatic initialization is beneficial as manual table creation is both time-consuming and highly error-plausible.

The values used for shuffling (referred to as the '*shuffle source'*) can be taken from one of several possible locations. Table 1 explains how these values can be picked from these locations. The location of the *shuffle source* that is used is recorded in the metadata of the attribute-value in the first 4 bits of a 7-bit string, as described in Table 1. For example, suppose the value *'The Coca-Cola Company'* in the *'Manufacturer'* key in a table in the *'Food'* category is replaced with the value *'Hood River Distillers'* which is a value in the *'Distributor'* key of a table in the *'Food'* category of the external split. Then, *'Hood River Distillers'* will have the metadata *'1011000'* after initialization[4]. These bits can be used to determine which way of shuffling was more effective, i.e., generate counterfactual data which have a greater impact on model performance, as demonstrated in the performance analysis (Section 4). The last 3 bits are explained in Section 3.2.

The initialization for the hypotheses and their labels is done by copying them exactly from the original dataset, and they are modified by human annotators using the TabPert platform.

Bit	Location	Same	Different
1	Split	0	1
2	Category	0	1
3	Table	0	1
4	Key	0	1

Table 1: First Four Bits of Table Value Metadata. These bits represent the location of the *shuffle source*. The 1st bit indicates whether is an external set (1) or the same set (0), the 2nd bit indicates whether it is a different (1) or the same (0) table category, the 3rd bit represents if it is the same (0) or a different (1) table, and the 4th represents whether it is the same (0) or a different (1) key. For values that do not change, the initial four bits are '0000'. Also, when the 3rd bit is 0, then the first two bits are necessarily 0.

3.2 Modifying Tables

Annotators can now modify these automatically-perturbed tables from initialization to remove self-contradictions and inconsistency to create valid counterfactual examples. All the cells (attributes and values) in the three counterfactual tables (A, B, and C) can be edited[5]. Table rows can be modified via the dragging and dropping of a value cell from (a) same counterfactual table (cut-paste effect), (b) from another counterfactual table (cut-paste effect), or (c) from the original table (copy-paste effect). To minimise errors during this drag-and-drop operation, a type validation check runs in the background, which prevents drag and drop between different key categories (for example, it is forbidden to drag a *Person's Name* into *Date of Birth*). To achieve such type validation, key 'entity type' must be provided before beginning the annotation procedure[6]. Keys for which this information is missing can be dropped anywhere without restriction.

TabPert also supports five additional functions for more challenging edits. The *'Add Value'* box allows annotators to add new values by dragging and dropping a new cell to the correct location and inputting the desired new value. Additionally, one can utilise the *'Add Section'* button for inserting an entirely new row. For deleting a value, drag and drop the desired cell to the *'Delete Vaue'* Box. A complete row can also be deleted with the *'Edit Section Name or Delete Section'* option. To edit the text, 'click' on the value and then edit it. These modification details are recorded automatically in the last 3 bits of the 7-bit metadata: the 5^{th} bit represents a copy-paste from the original table, the

[2] https://github.com/utahnlp/tabpert

[3] Only a subset of all values are shuffled at random. The location of the shuffle source (described later) is likewise picked at random among this subset.

[4] Values taken from different keys, must have the same entity-type, as explained in Section 3.2.

[5] The original table cannot be changed. This is done to prevent inadvertent edits.

[6] This can be done manually or using NER tagging.

6^{th} represents a value update operation, and the 7^{th} bit represents a new cell or row addition.

Figure 3a shows the main parts of the TABPERT platform for counterfactual table perturbation.

3.3 Hypothesis Modification and Metadata

The text of a hypothesis of a counterfactual table can be edited directly, and its corresponding label can also be selected from drop-down menu options. In addition, the annotator enters the following metadata information:

1. The strategies used by the annotator to modify the hypothesis. The five main strategies can be selected using check-boxes (selecting multiple values is allowed), as shown in Figure 3c. The *'Other'* option corresponds to hypothesis changes that do not fall into the five main strategies.

2. All the relevant rows of the table which are necessary for deciding the inference label.

Figure 3b shows the main TABPERT view for hypothesis modification, with hypothesis and inference label. The annotator inserts metadata by clicking the '+' symbol on the left side of each counterfactual hypothesis (below label drop-down), as shown in Figure 3b. This opens a metadata collection window, as shown in Figure 3c. We use 6 bits to store this metadata information: each of the initial 5 bits represents a strategy (the order of the bits is the same as the order in which the strategies are mentioned on TABPERT as shown in Figure 3c). The last bit represents the 'Other' option. Additionally, the relevant rows' 'attribute keys' are stored in a list (array) along with each modified hypothesis.

3.4 TABPERT Aspects

The TABPERT web-app's core tech stack consists of ReactJS[7] and Flask[8]. Here, Flask is used as the main back-end Python web framework, and Javascript library ReactJS is used for the front-end. We used Flask because it is easy to extend, giving us the ability to easily integrate Python libraries to manipulate JSON and TSV files quickly. We used ReactJS because of the *react-beautiful-dnd* library[9] essential for simulating the drag-and-drop function.

4 Case Study on INFOTABS

We used TABPERT to create counterfactual data for the INFOTABS dataset. Each table is saved as a JSON file with keys and values as attributes in INFOTABS. We sampled 47 tables with 423 table-hypothesis pairs taken from the α_1 set of INFOTABS. For initialization, we shuffled the entities in this sampled α_1 set with those in the tables from both the $Train$ set (the 'external' set) and the complete α_1 set (the 'internal' set). Including both sets creates more diversity in automatic initialization[10].

Annotation Guidelines Following a similar line as earlier works by Ribeiro et al. (2020b) and Sakaguchi et al. (2020) for creating challenging adversarial test sets, we guided the annotators in annotating three counterfactual tables (A, B, C) for each original α_1 table. This ensures enough diversity and coverage in the collected counterfactual data. For each counterfactual table, we encouraged annotators to use the following strategies: (a) For *Table A*: change the table such that the *entailment* (E) and *contradiction* (C) labels are flipped, but the hypothesis remains unchanged, (b) For *Table B*: change the hypothesis so that the *entailment* (E) and *contradiction* (C) labels are flipped; make any necessary changes to the table, and (c) For *Table C*: write a new but related hypothesis with similar reasoning; the table can be modified as needed. We also recommend that annotators modify the *neutrals* (N) by adding more *'true'* information from the table to the hypotheses to make them more challenging. The above-discussed procedure ensures that (a) the final labels are balanced, (b) the reversed label eliminates hypothesis bias (Gupta et al., 2020; Chen et al., 2019), and (c) due to lexical overlap, *neutrals* (N) are closer to *entailments* (E) (Glockner et al., 2018). Finally, after annotation, we have 109 counterfactual tables with a total of 982 table-hypothesis pairs, with Table A having 423, Table B having 405, and Table C having 154 pairs.

Experiment and Analysis To check if the annotated counterfactual data is challenging for existing models, we use RoBERTa$_{Large}$ to obtain prediction labels for the original and counterfactual data. We also obtain hypothesis-only baseline predictions using the RoBERTa$_{Large}$ model on the two sets. Table 2 shows the performance results in the form

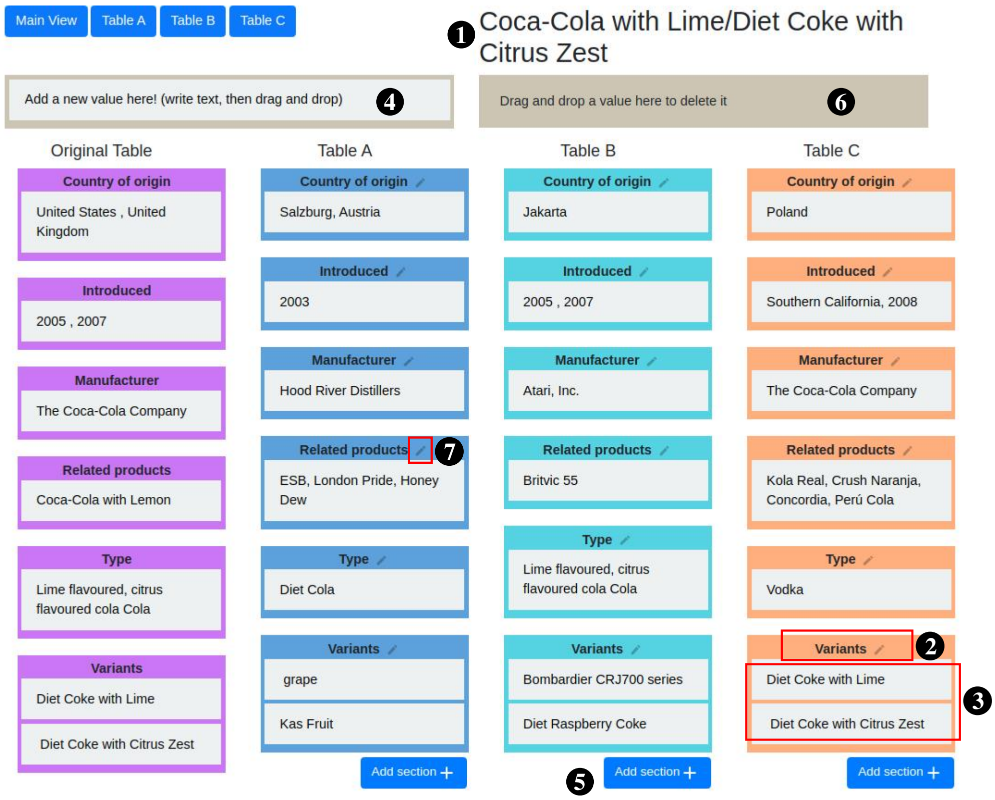
[7] https://reactjs.org/

[8] https://flask.palletsprojects.com/en/2.0.x/

[9] https://www.npmjs.com/package/react-beautiful-dnd

[10] As discussed earlier, annotators manually fix the constraint violations during annotation.

(a) **Table Perturbation:** ❶ Table Title ❷ a Key (Section Name) ❸ the Values associated with a Key ❹ Add Value ❺ Add Section ❻ Delete Value ❼ Edit Section Name or Delete Section

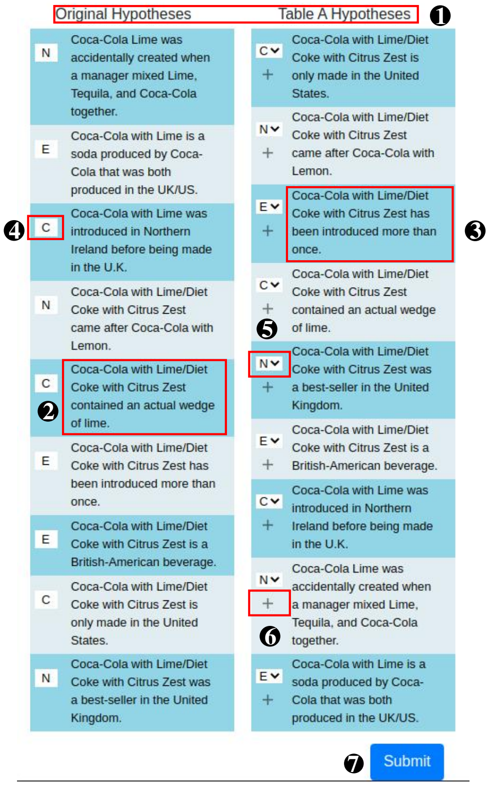


Coca-Cola with Lime/Diet Coke with Citrus Zest

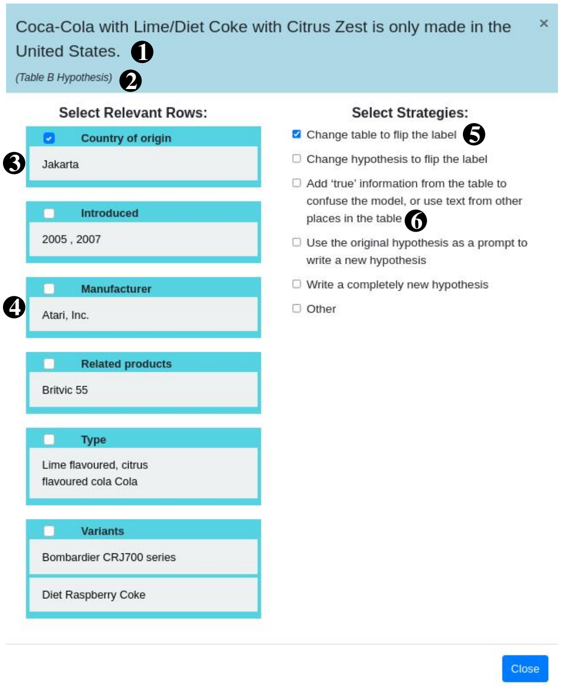
Add a new value here! (write text, then drag and drop)

Drag and drop a value here to delete it

	Original Table	Table A	Table B	Table C
Country of origin	United States , United Kingdom	Salzburg, Austria	Jakarta	Poland
Introduced	2005 , 2007	2003	2005 , 2007	Southern California, 2008
Manufacturer	The Coca-Cola Company	Hood River Distillers	Atari, Inc.	The Coca-Cola Company
Related products	Coca-Cola with Lemon	ESB, London Pride, Honey Dew	Britvic 55	Kola Real, Crush Naranja, Concordia, Perú Cola
Type	Lime flavoured, citrus flavoured cola Cola	Diet Cola	Lime flavoured, citrus flavoured cola Cola	Vodka
Variants	Diet Coke with Lime Diet Coke with Citrus Zest	grape Kas Fruit	Bombardier CRJ700 series Diet Raspberry Coke	Diet Coke with Lime Diet Coke with Citrus Zest



	Original Hypotheses		Table A Hypotheses
N	Coca-Cola Lime was accidentally created when a manager mixed Lime, Tequila, and Coca-Cola together.	C	Coca-Cola with Lime/Diet Coke with Citrus Zest is only made in the United States.
E	Coca-Cola with Lime is a soda produced by Coca-Cola that was both produced in the UK/US.	N	Coca-Cola with Lime/Diet Coke with Citrus Zest came after Coca-Cola with Lemon.
C	Coca-Cola with Lime was introduced in Northern Ireland before being made in the U.K.	E	Coca-Cola with Lime/Diet Coke with Citrus Zest has been introduced more than once.
N	Coca-Cola with Lime/Diet Coke with Citrus Zest came after Coca-Cola with Lemon.	C	Coca-Cola with Lime/Diet Coke with Citrus Zest contained an actual wedge of lime.
C	Coca-Cola with Lime/Diet Coke with Citrus Zest contained an actual wedge of lime.	N	Coca-Cola with Lime/Diet Coke with Citrus Zest was a best-seller in the United Kingdom.
E	Coca-Cola with Lime/Diet Coke with Citrus Zest has been introduced more than once.	E	Coca-Cola with Lime/Diet Coke with Citrus Zest is a British-American beverage.
E	Coca-Cola with Lime/Diet Coke with Citrus Zest is a British-American beverage.	C	Coca-Cola with Lime was introduced in Northern Ireland before being made in the U.K.
C	Coca-Cola with Lime/Diet Coke with Citrus Zest is only made in the United States.	N	Coca-Cola Lime was accidentally created when a manager mixed Lime, Tequila, and Coca-Cola together.
N	Coca-Cola with Lime/Diet Coke with Citrus Zest was a best-seller in the United Kingdom.	E	Coca-Cola with Lime is a soda produced by Coca-Cola that was both produced in the UK/US.



Coca-Cola with Lime/Diet Coke with Citrus Zest is only made in the United States.

(Table B Hypothesis)

Select Relevant Rows:

- ☑ Country of origin: Jakarta
- □ Introduced: 2005 , 2007
- □ Manufacturer: Atari, Inc.
- □ Related products: Britvic 55
- □ Type: Lime flavoured, citrus flavoured cola Cola
- □ Variants: Bombardier CRJ700 series; Diet Raspberry Coke

Select Strategies:

- ☑ Change table to flip the label
- □ Change hypothesis to flip the label
- □ Add 'true' information from the table to confuse the model, or use text from other places in the table
- □ Use the original hypothesis as a prompt to write a new hypothesis
- □ Write a completely new hypothesis
- □ Other

(b) **Hypotheses Perturbation:** ❶ Tables corresponding to the hypothesis sets ❷ an Original Hypothesis ❸ a Counterfactual Hypothesis of Table A ❹ the NLI Label corresponding to an Original Hypothesis ❺ the NLI Label corresponding to a Table A Counterfactual Hypothesis ❻ Open Modal for Hypothesis Metadata (Figure 3c) ❼ Save Option

(c) **Hypothesis Metadata:** (Select Relevant Rows and Hypothesis Perturbation Strategies) ❶ the Hypothesis ❷ Table Name ❸ a Relevant Row (checkbox selected) ❹ an Irrelevant Row (checkbox unselected) ❺ a Used Strategy (checkbox selected) ❻ an Unused Strategy (checkbox unselected)

Figure 2: Main Features of the TABPERT Platform

of accuracy. The table-sentence data was represented in *'para'* form (Gupta et al., 2020) in two ways: (a) with all table rows, (b) using only the relevant rows (obtained via annotated metadata) (Gupta et al., 2021).

Performance Analysis It is evident from Table 2 that RoBERTa$_{Large}$ has difficulty in predicting labels correctly for counterfactual data. Furthermore, the model's higher performance with relevant rows indicates that it most likely utilises irrelevant rows as artefacts when making predictions (Neeraja et al., 2021). On counterfactual data, the hypothesis-only model's performance is close to majority-label baselines, confirming a reduction in hypothesis bias. Humans, on the other hand, find both datasets equally difficult and obtain an accuracy of $\approx 85\%$ on each[11].

Model Type	Original	Counterfactual
Majority	33.33	33.33
Hypo Only	64.32	44.85
All Rows	78.91	61.26
Relevant Rows	74.11	65.85
Human	84.8	85.8

Table 2: Performance (accuracy %) of the INFOTABS RoBERTa$_{Large}$ model on original and counterfactual annotated data.

Perturbation Analysis We also study the hypothesis annotation metadata to see which hypothesis modification strategies are more effective. From Figure 4, it is evident that manual *Table Change* with *Label Flip* (*TC + LF*) is more effective than manual *Hypothesis Change* with *Label Flip* (*HC + LF*). Furthermore, all *Label Flip* methods are typically more effective than *Hypothesis Prompt* (*HypoPrompt*) and *Text Overlap* (*Overlap*). This, we believe, is due to the ineffectiveness of hypothesis bias with flipped labels. Surprisingly, there is a modest performance increase on new hypotheses, showing that simple data generation is an unsuccessful method. Furthermore, no *Other* perturbation techniques result in any substantial performance drop.

We also did a similar analysis on the table perturbation metadata; refer to Section A of the Appendix for details. Refer to Section C of the Appendix examples of counterfactual perturbation using each strategy.

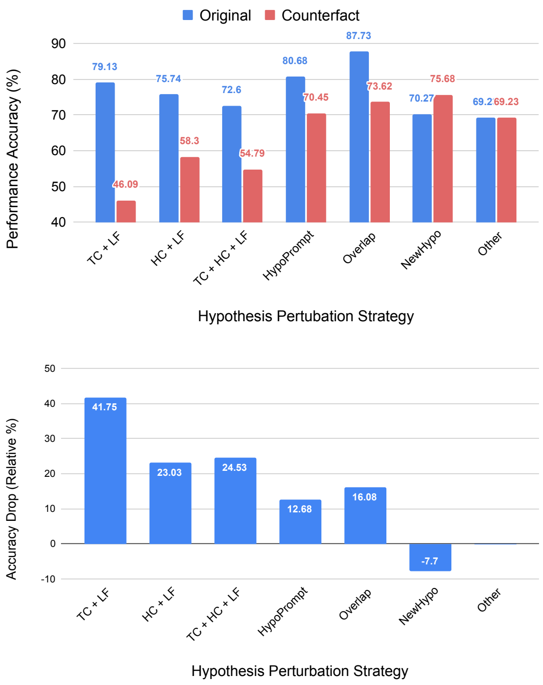


Figure 4: Performance drop after counterfactual perturbation with various strategies.

5 TABPERT Utility

Main Platform TABPERT is a tool designed for efficient and accurate table perturbation. One such case is creating tabular data for tabular inference tasks, as demonstrated through this paper. TABPERT supports several features which aid in the creation of effective counterfactual tabular data. It has numerous optimizations with a friendly user interface to ensure fast annotation of data. This ensures huge data collection, leading to scalability. TABPERT enables a larger range of services compared to using spreadsheets or the MTurk platform. For example, the drag-and-drop functionality simplifies annotation and helps easily visualise a complicated job. All the tabular data can be examined in a single view. The automatic type validation during initialisation and modification reduces the chances of unintended errors.

Customizing TABPERT Functionality The initialization source code, as well as the platform, are designed to be modular. This facilitates component addition, deletion, and updation. For example, the ability to reorganize table parts, copy values across table triplets (in addition to cut-paste), auto-save work[12], and an undo option, as well as checkpoints

[11] There is no difference in performance between A, B, and C type counterfactual table-example pairs. See Figure 5 in the Appendix.

[12] Currently, the Save button must be pressed manually to save work. However, even semi-completed work can be saved

added to reverse mistakes, may all be readily implemented. The augmentation initialization code can also be configured to suit the requirements of the task.

Metadata Counterfactual data can be utilized as a difficult adversarial test set to assess tabular model reasoning. For example, as demonstrated in Section 4, The model performs poorly on counterfactual hypotheses with flipped labels on tables with relevant rows drawn from the external set (in our case, the $Train$ set), indicating model overfitting on pre-trained knowledge. The recorded metadata can also be utilized to guide annotators in creating increasingly difficult data. For example, annotators can be encouraged to focus more on *Label Flip* methods with external set-initialised tables rows to generate more challenging counterfactual data. Label flipping techniques can also be used to test a model for hypothesis bias. The metadata associated with hypotheses-specific relevant rows assists in pruning premise tables, which improves inference model reasoning and interpretability, as shown in Section 4. These are only a handful of the countless possible application scenarios.

In Section B of the Appendix, we also compare and contrast TABPERT with spreadsheets on effectiveness, visual benefits, and metadata aspects.

6 TABPERT Limitations and Future

During our pilot study, the platform was run locally by three annotators. This was not an issue because the number of annotators was limited, and the tables were divided among them manually. If we want numerous annotators to be able to make simultaneous modifications for large-scale distribution, we must host our platform on a centralized server. This is something we intend to accomplish in the not-too-distant future.

Finally, the counterfactual data generated by modifications has to be manually stored by pressing a button. This was done so that if the user made a mistake, the original data would not be erased, and the user may save the data after they are satisfied with the modifications. To accommodate both of these circumstances, we would like to include an auto-save function along with an undo option.

7 Conclusions

TABPERT is an effective platform for examining semi-structured tabular data and generating counterfactual tabular perturbations. Annotators can use the platform to alter tables and hypothesis phrases, as well as collect related metadata information, in order to produce tabular counterfactual data. The metadata collected can be utilised to analyze the unknown vulnerabilities with existing NLP systems quantitatively and methodically. We believe that TABPERT will be helpful to academics that work with semi-structured data such as tables. Many non-academic industrial scenarios that require table modification, such as e-commerce product specification tables, financial and tax statements tables, and so on, may also leverage TABPERT.

8 Acknowledgements

We appreciate the Utah NLP group members' valuable suggestions at various phases of the project, as well as the reviewers' helpful remarks. We also acknowledge NSF grants #1801446 (SATC) and #1822877 (Cyberlearning), as well as a kind donation from Verisk Inc. We also like to thank Alokit Innovations for providing a mentoring platform.

References

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*.

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

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. TabFact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

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.

by pressing the button several times at regular intervals.

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.

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*.

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.

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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020a. Beyond Accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020b. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WINOGRANDE: An adversarial winograd schema challenge at scale. In *AAAI*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Nancy XR Wang, Diwakar Mahajan, Marina Danilevsk Rosenthal, et al. 2021. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). *arXiv preprint arXiv:2105.13995*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix: Performance vs Perturbation

Figure 5 shows the relative accuracy drop of the model performance for each table perturbation strategy. There is no significant difference in the average accuracy across the A, B, and C counterfactual types. Figure 6 shows the number of examples for each hypothesis perturbation strategy. *Label Flip* (*LF*) is frequently used by the annotators with either the hypothesis or the table changes. Annotators also regularly use the *HypoPrompt* and Hypothesis overlap strategy for creating counterfactuals. Annotators avoid making new hypotheses.

B Appendix: TabPert vs Spreadsheets

Effectiveness When utilizing spreadsheets for annotation, it becomes quite difficult and time-consuming to cut/copy-paste cells. The efficient drag-drop feature with automatic type restrictions in TabPert makes it a much easier and faster procedure. Editing and altering text in TabPert is also easier compare to that on a spreadsheet. Our study found that it takes around 7 minutes on average to annotate a new table with 9 statements using TabPert, but the same work done using a spreadsheet takes more than 30 minutes.

Visualization Benefits TabPert's table visualisation provides a view of the entire data on a single screen. Seeing the entire picture (tables and hypotheses) is incredibly helpful for assessing the quality of annotations. It also allows the

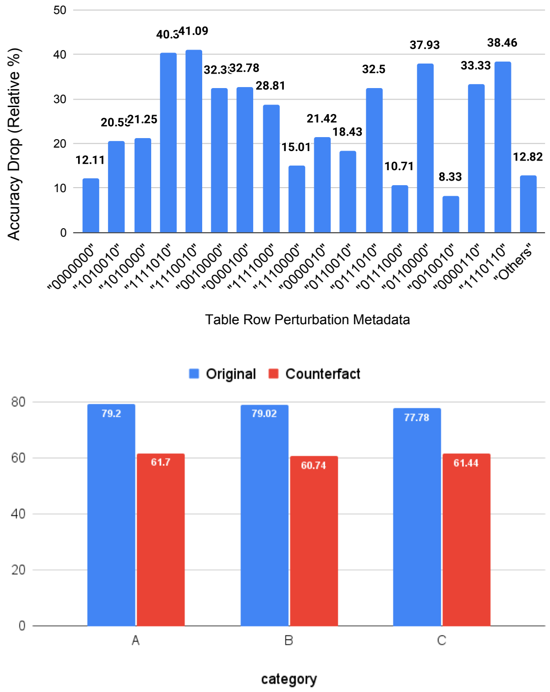
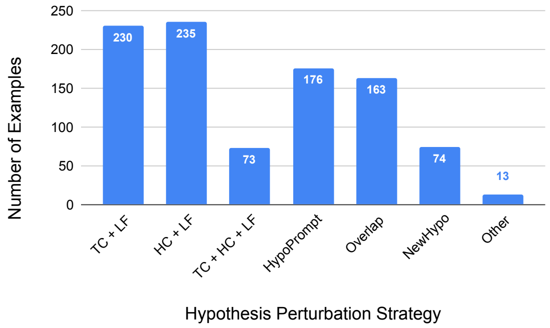


Figure 5: (Top) Performance drop with counterfactual perturbation with several perturbation strategies (using Table 1 for interpreting the analysis). (Bottom) Performance on A,B and C counterfactual tables.



si

Figure 6: Number of examples of each hypothesis perturbation strategy.

annotator to quickly follow label and hypothesis changes, which is not feasible in a cumbersome spreadsheet's view.

Furthermore, having a single screen 'Focus View' on a single counterfactual table makes altering hypotheses even easier. Using this focus feature, updating the labels or adding new information to the hypothesis is straightforward. This focus view is not viable with a spreadsheet; to make appropriate alterations, one must search and navigate to each spreadsheet cell.

In addition to this, the lack of scrolling required while dragging and dropping on our platform saves annotators time. To discover the relevant cells in a spreadsheet, one must execute numerous scrolling operations to the up, down, left, or right.

Finally, in TABPERT, the cell size is set to exactly fit its contents, but in a spreadsheet, cells in each row and columns have the same height and width, making it quite problematic to view text properly.

Metadata Collection TABPERT makes it simple to gather information such as methods used to change a hypothesis and rows utilized to answer each hypothesis, using checkboxes. In a spreadsheet, this would require 9 columns of checkboxes for each table or manually writing the metadata, which is now automatically done with a single click, thus making the process simple and efficient. Moreover, automatic metadata collection about a drag and drop location is not possible in a spreadsheet.

C Appendix: Qualitative Counterfactual Perturbation Examples

Tables 3, 4, 5, 6, 7 illustrate the five strategies used for counterfactual table-hypothesis perturbation. Here, *Before* and *After* row represent hypothesis and corresponding relevant table rows[13] before and after counterfactual perturbation. In the *After* row, we also provide the 7-bit meta-data associated with each row value. Finally, the *Automatic Initialisation* row explains the meaning of the first four bits of this meta-data, and the *Manual Editing* row explains the last three bits, for all the value in concern.

[13] For simplicity, we only include the rows of the table relevant to the hypotheses.

	Premise	Hypothesis	Label	Predicted
Before (T14)	Box Office 1. $61.3 million Budget 1. $26 million	Flatliners made over double what it cost to make at the box office.	E	E
After (T14A)	Box Office 1. $ 140.7 million (1010 010) Budget 1. $85 million (0111 010)	Flatliners made over double what it cost to make at the box office.	C	E
Automatic Initialisation	1010: different dataset, same category, different table, same key 0111: same dataset, different category, different table, different key			
Manual Editing	010: value text edited			

Table 3: Example using Strategy 1
Strategy: Change table to flip label (TC+LF)

	Premise	Hypothesis	Label	Predicted
Before (T14)	Box Office 1. $61.3 million Budget 1. $26 million	Flatliners made over double what it cost to make at the box office	E	E
After (T14B)	Box Office 1. $ 13.3 million (1010 010) Budget 1. $5.9 million (0110 010)	Flatliners made over triple what it cost to make at the box office.	C	E
Automatic Initialisation	0110: same dataset, different category, different table, same key 1010: different dataset, same category, different table, same key			
Manual Editing	010: value text edited			

Table 4: Example using Strategy 2
Strategy: Change hypothesis to flip label (HC+LF)

	Premise	Hypothesis	Label	Predicted
Before (T14)	Produced by 1. Michael Douglas 2. Rick Bieber Directed by 1. Joel Schumacher	Rick Bieber put more money into Flatliners than Michael Douglas did.	N	N
After (T14A)	Produced by 1. Rick Bieber (0000 100) 2. Michael Douglas (0000 100) Directed by 1. Empress Teimei (1111 000)	Rick Bieber put more money into Flatliners directed by Empress Teimei, than Michael Douglas did	N	N
Automatic Initialisation	0000: same dataset, same category, same table, same key 1111: different dataset, different category, different table, different key			
Manual Editing	000: no change 100: copied from the original table			

Table 5: Example using Strategy 3
Strategy: Add 'true' information from the table to confuse the model (Overlap)

	Premise	Hypothesis	Label	Predicted
Before (T14)	Edited by 1. Robert Brown Written by 1. Peter Filardi	Flatliners was Peter Filardi's first writing credit.	N	N
After (T14B)	Edited by 1. James Newton Howard (0000 100) 2. Robert Brown (0000 000) Written by 1. Lee Beom-seon (1110 000)	Flatliners was mostly edited by Robert Brown.	N	E.
Automatic Initialisation	0000: same dataset, same category, same table, same key 1110: different dataset, different category, different table, same key			
Manual Editing	000: no change 100: copied from the original table			

Table 6: Example using Strategy 4
Strategy: Use the original hypothesis to write a new hypothesis (HypoPrompt)

	Premise	Hypothesis	Label	Predicted
Before (T14)	Box Office 1. $61.3 million Budget 1. $26 million	Flatliners made 50 million over it's budget at the box office.	C	E
After (T14C)	Box Office 1. US$85.4 million (December 2017) (0111 000) Budget 1. $26 million (0000 000)	Flatliners costed around $25 million in making and was a hit.	E	C
Automatic Initialisation	0000: same dataset, same category, same table, same key 0111: same dataset, different category, different table, different key			
Manual Editing	000 : no change			

Table 7: Example using Strategy 5
Strategy: Write a completely new hypothesis (NewHypo)