OpenRT: An Open-source Framework for Reasoning Over Tables

Yilun Zhao[*1] **Boyu Mi**[*2] **Zhenting Qi**[2] **Linyong Nan**[1]
Minghao Guo[2] **Arman Cohan**[1] **Dragomir Radev**[1]
[1]Yale University [2] Zhejiang University
yilun.zhao@yale.edu, miboyu@zju.edu.cn

Abstract

There are a growing number of table pre-training methods proposed for reasoning over tabular data (e.g., question answering, fact checking, and faithful text generation). However, most existing methods are benchmarked solely on a limited number of datasets, varying in configuration, which leads to a lack of unified, standardized, fair, and comprehensive comparison between methods. This paper presents OpenRT, the first open-source framework for reasoning over tabular data, to reproduce existing table pre-training models for performance comparison and develop new models quickly. We implemented and compared six table pre-training models on four question answering, one fact checking, and one faithful text generation datasets. Moreover, to enable the community to easily construct new table reasoning datasets, we developed TaRAT, an annotation tool which supports multi-person collaborative annotations for various kinds of table reasoning tasks. The researchers are able to deploy the newly-constructed dataset to OpenRT and compare the performances of different baseline systems. The library OpenRT, along with the annotation tool TaRAT, is publicly available at https://github.com/yilunzhao/OpenRT.

1 Introduction

With the increasing amount of structured data available, there is a growing interest in developing NLP systems for reasoning over tabular data to perform tasks such as question answering (Pasupat and Liang, 2015; Zhong et al., 2017; Iyyer et al., 2017), fact checking (Chen et al., 2020c; Gupta et al., 2020), and faithful text generation (Chen et al., 2020b; Parikh et al., 2020). Table pre-training has emerged as a promising approach for developing large language models (LLMs) that can perform various kinds of downstream table reasoning tasks with high accuracy after fine-tuning (Herzig et al., 2020; Liu et al., 2022b; Jiang et al., 2022; Yang et al., 2022; Zhao et al., 2022b; Liu et al., 2022a). However, existing table pre-training methods have been benchmarked on different datasets with varying configurations (Table 2), resulting in a lack of standardization for comprehensive evaluation between methods. Moreover, existing models are developed under individual systems and have a lack of compatibility. Therefore, it is difficult and time-consuming to re-implement them for result comparison in future studies. As the above issues seriously hinder the development of table reasoning models, it is imperative to develop a unified and extensible open-source framework for reasoning over tabular data.

In this paper, we present **OpenRT**, the first **Open**-source framework for **R**easoning over **T**abular data, which has the following three characteristics: (1) *Modularization*: we developed OpenRT with highly reusable modules and integrated them in a unified framework, which enables researchers to study different table reasoning models at a conceptual level; (2) *Standardization*: OpenRT includes popular table reasoning datasets and models. The evaluation of different models is standardized under the same experimental configuration; (3) *Extensibility*: OpenRT enables researchers to easily develop their own models or add new datasets by extending corresponding modules with their proposed ones.

Moreover, in order to facilitate the construction of new table reasoning datasets by other researchers, we developed **TaRAT**, the first **TA**ble **R**easoning **A**nnotation **T**ool that supports the collaborative construction of various dataset types (i.e., question answering, fact checking, text generation). User-created datasets can be easily integrated into OpenRT for performance evaluation.

The main structure of the paper is organized as follows: Section 2 describes each table reason-

*Equal Contributions.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 336–347
July 10-12, 2023 ©2023 Association for Computational Linguistics

Dataset	# Examples	# Tables	Input	Output	Evaluation Metrics
			Question Answering		
WikiSQL (Zhong et al., 2017)	80,654	24,241	question	short-form answer	Acc
WTQ (Pasupat and Liang, 2015)	22,033	2,108	question	short-form answer	Acc
SQA (Iyyer et al., 2017)	17,553	982	sequential question	sequential answers	Acc
FeTaQA (Nan et al., 2022a)	10,330	10,330	question	long-form answer	B, R, BS, PARENT, NLI-Acc
			Fact Checking		
TabFact (Chen et al., 2020c)	118,275	16,573	statement	entailment label	Acc
			Faithful Table-to-Text Generation		
LogicNLG (Chen et al., 2020a)	37,015	7,392	highlighted columns	statement	B, R, BS, PARENT, SP/NLI-Acc

Table 1: Table reasoning tasks in OpenRT. B denotes BLEU, R denotes ROUGE, and BS denotes BERTScore. The details of each evaluation metric are introduced in Appendix A.

	WikiSQL	WTQ	SQA	FeTaQA	TabFact	LogicNLG
TaPas (Herzig et al., 2020)	✓	✓	✓		✓	
UnifiedSKG (Xie et al., 2022)	✓	✓	✓	✓	✓	✓
TAPEX (Liu et al., 2022b)	✓	✓	✓	✓	✓	✓
ReasTAP (Zhao et al., 2022b)	✓	✓	✓	✓	✓	✓
OmniTab (Jiang et al., 2022)	✓	✓	✓	✓	✓	✓
PLOG (Liu et al., 2022a)	✓	✓	✓	✓	✓	✓

Table 2: The list of table reasoning datasets used in different table pre-training works. It demonstrates the lack of standardized and comprehensive benchmarks for evaluating existing table pre-training methods.

ing task included in OpenRT; Section 3 describes each module and its implementation of OpenRT framework; Section 4 compares the performance of different table pre-training methods on included datasets, and provides insights into how to choose appropriate table pre-training methods for specific needs; Section 5 introduces the functions and implementation of TaRAT; finally, Section 6 introduces the related work about table reasoning and annotation tools.

2 OpenRT Tasks

OpenRT covers three kinds of table reasoning tasks: question answering, fact checking, and faithful text generation. The goal of OpenRT is to push the development of table pre-training methods that can be applied and achieved competitive performance on various kinds of table reasoning tasks. We describe the details of each dataset in the following subsections and Table 2.

2.1 Table Question Answering

WikiSQL The WikiSQL-Weak dataset (Zhong et al., 2017) requires models to perform filtering and, optionally, aggregation on table cell values to obtain an answer to the given question.

WTQ The WikiTableQuestions dataset (Pasupat and Liang, 2015) contains 22,033 complex questions on Wikipedia tables. Compared to WikiSQL, it requires more complicated reasoning capabilities, thus is more challenging.

SQA The SequentialQA dataset (Iyyer et al., 2017) was built by decomposing the questions from WTQ dataset and organizing them into a conversational context. It requires models to answer sequences of simple but interrelated questions.

FeTaQA Different from above-mentioned three *short-form* Table QA datasets, the Free-form Table Question Answering dataset (Nan et al., 2022b) requires models to generate *free-form* text answers after retrieval, inference, and integration of multiple supporting facts from the source table.

2.2 Table Fact Checking

TabFact The TabFact dataset (Chen et al., 2020c) requires the models to perform both soft linguistic reasoning and hard symbolic reasoning to determine whether a given statement is entailed or refuted by the corresponding tabular data.

2.3 Faithful Table-to-Text Generation

LogicNLG The LogicNLG dataset (Chen et al., 2020a) requires models to generate multiple statements that perform logical reasoning based on the information in the source table. Each statement

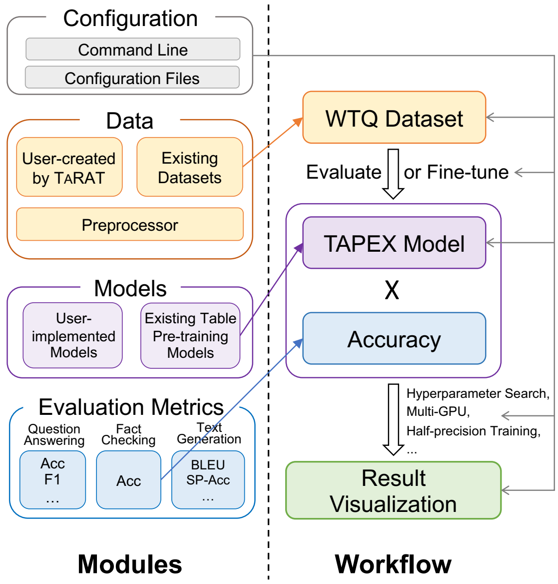


Figure 1: The overall framework of OPENRT.

should be factually correct with the table content.

3 OPENRT Framework

As shown in Figure 1, OPENRT consists of four main modules: configuration, data, modeling, and evaluation. The users are able to fine-tune or test the existing table pre-training models on the included dataset. They are also allowed to add their own models or datasets into OPENRT by extending corresponding modules with their proposed ones.

3.1 Configuration Module

Users and developers define all experiment configurations in the configuration module, which includes command lines, external configuration, and internal configuration. Users are expected to modify the major experiment settings through command lines or by modifying external configuration files, while keeping the internal configuration unchanged for replicating existing models. This ensure a unified and standardized performance comparison between different table reasoning models.

3.2 Data Module

As discussed in Section 2, OPENRT includes popular datasets for table reasoning, which cover various types of tasks. Any raw dataset undergoes processing using the following data flow: raw data → *Preprocessor* → *Dataset* → *Dataloader* → processed data. The data flow converts raw datasets in various formats into a unified format that can be used as input for the modeling module.

The *Preprocessor* tokenizes textual and tabular data input using the corresponding tokenizer of the model. It applies the same strategy as Liu et al. (2022b) to truncate a long table into a shorter version to satisfy the model's input length limit. The *Dataset* component prepares input data, while the *DataLoader* component selects features from the processed data to form tensor data for model input. For both components, we have implemented parent classes `TRDataset` and `TRDataLoader` to include shared attributes and functions. Users can add a new dataset by creating classes that inherit from these parent classes with a few modifications.

3.3 Modeling Module

We have organized and unified the implementations of each table reasoning model within the modeling module by creating an interface parent class called `TRModel`. The design of `TRModel` simplifies the process for users who want to deploy or add a new model to OPENRT. They can simply create and modify a corresponding child class by inherit `TRModel`. The following table reasoning models have been implemented in OPENRT:

- **TAPAS** (Herzig et al., 2020) adopts the BERT encoder with an additional positional embedding for encoding table structure. It also adds two classification layers for cell selection and aggregation operator predictions.
- **UnifiedSKG** (Xie et al., 2022) unifies each task into a text-to-text format, and adopts a sequence-to-sequence T5 model for multi-task learning over multiple table reasoning datasets.
- **TAPEX** (Liu et al., 2022b) pre-trains LLMs by learning as a neural SQL executor to predict the execution results of synthetic SQL queries.
- **REASTAP** (Zhao et al., 2022b) injects various kinds of table reasoning skills (e.g., conjunction, counting) into LLMs by synthesizing Table QA examples as the pre-training corpus.
- **OmniTab** (Jiang et al., 2022) retrieves table-sentence pairs from Wikipedia for mask-based pre-training and synthesizes Table QA examples for pre-training with a QA loss.
- **PLOG** (Liu et al., 2022a) is pre-trained on a synthetic corpus of table-to-logic-form generation to learn table-relevant logical inference knowledge.

While it is possible to train a single model for each task without using the "pre-train, then fine-tune" paradigm (Zhou et al., 2022; Ou and Liu,

	FeTaQA				LogicNLG					
	B-4	ROUGE-1/2/L	BS	NLI	BLEU-1/2/3	ROUGE-1/2/L	BS	PA	SP	NLI
UnifiedSKG	31.5	63.5/41.8/54.1	83.6	78.0	51.8/32.5/18.8	42.8/20.9/36.5	75.1	32.9	46.2	87.0
TAPEX	30.2	62.0/39.9/50.7	82.3	79.2	52.2/32.1/18.3	44.0/21.5/36.8	72.5	31.9	50.1	87.4
ReasTAP	30.4	62.5/40.3/51.1	82.7	80.4	52.5/32.5/18.9	44.2/21.5/37.3	78.2	32.2	54.8	**89.2**
OmniTab	30.7	62.9/40.6/52.1	84.1	**81.5**	53.0/32.9/19.1	44.5/21.7/37.4	77.6	31.7	**55.1**	89.0
PLOG	**31.8**	**64.7/42.5/54.9**	**86.2**	80.2	**54.9/35.0/21.0**	**46.1/23.8/39.0**	**80.1**	**32.8**	50.5	88.9

Table 3: Automated Evaluation of table pre-training models on the test set of FeTaQA and LogicNLG datasets. BS denotes BERTScore, PA denotes PARENT, SP denotes SP-Acc, and NLI denotes NLI-Acc.

	Short-form QA			Fact Checking
	WikiSQL	WTQ	SQA	TabFact
PLOG	85.9	43.7	60.3	82.0
UnifiedSKG	85.6	48.3	61.5	83.5
TaPas	84.0	50.4	67.1	81.0
TAPEX	<u>89.2</u>	57.2	74.5	84.0
ReasTAP	**90.4**	<u>58.6</u>	<u>74.7</u>	<u>84.7</u>
OmniTab	88.7	**62.8**	**75.9**	**85.2**

Table 4: Accuracies of existing table pre-training models on the test set of short-form table QA and table fact checking datasets. Bold numbers indicate the highest accuracy, and underscores denote the second best.

2022; Zhao et al., 2023a), we included only *table pre-training models* in OpenRT. This is because we focus on pushing forward the development of more generalizable table pre-training methods that can be applied to various table reasoning tasks and achieve competitive performance.

3.4 Evaluation Module

To evaluate and compare the performance of table reasoning models supported by a certain dataset, OpenRT includes all the evaluation metrics used in the official implementation. These metrics can be used off-the-shelf with a one-line call. The details of each metric are introduced in Appendix A.

3.5 Execution

We implemented *Evaluation* and *Fine-tuning* paradigms for execution in OpenRT. For *Evaluation*, users are able to replicate experimental results of existing models on the supported table reasoning dataset by using provided model checkpoints[1]. For *Fine-tuning*, they can train existing models on new datasets or fine-tune their self-implemented models on the included datasets. OpenRT supports hyper-parameter search to improve fine-tuning performance. We also implemented strategies such as multi-GPU training and half-precision training for efficient model training.

[1]We provide checkpoints of each supported model fine-tuned on each included dataset at https://huggingface.co/OpenTR

4 Experiments

4.1 Implementation Details

We conducted experiments to evaluate and compare the fine-tuning performance of supported table pre-training models on the included table reasoning datasets. In our experiments, if a model had been fine-tuned on a certain dataset in its original paper and its corresponding checkpoint was publicly available, we evaluated the model's performance directly using the provided checkpoint. Otherwise, we fine-tuned the model first and then evaluated its performance. For each fine-tuning experiment, we ran 40 epochs with a batch size of 128, and the best fine-tuning checkpoints were selected based on the validation loss.

4.2 Experimental Results

As shown in Table 3, PLOG achieves higher performance for most surface-level evaluations (i.e., BLEU, ROUGE, BERTScore, and PARENT) on faithful table-to-text generation and free-form Table QA tasks. This is reasonable because PLOG is pre-trained to generate logical forms given the tabular data, which improves the model's capability for content selection and logical inference in text generation. OmniTab achieves the best performance on faithfulness-level evaluation (i.e., SP-Acc and NLI-Acc). It also achieves the best performance on most fact checking and short-form QA tasks (Table 4), demonstrating the effectiveness of pre-training models over natural and synthetic Table QA examples to improve the model's reasoning capability. Our aim is that such performance comparison, using a standardized benchmark, will provide researchers with valuable insights on how

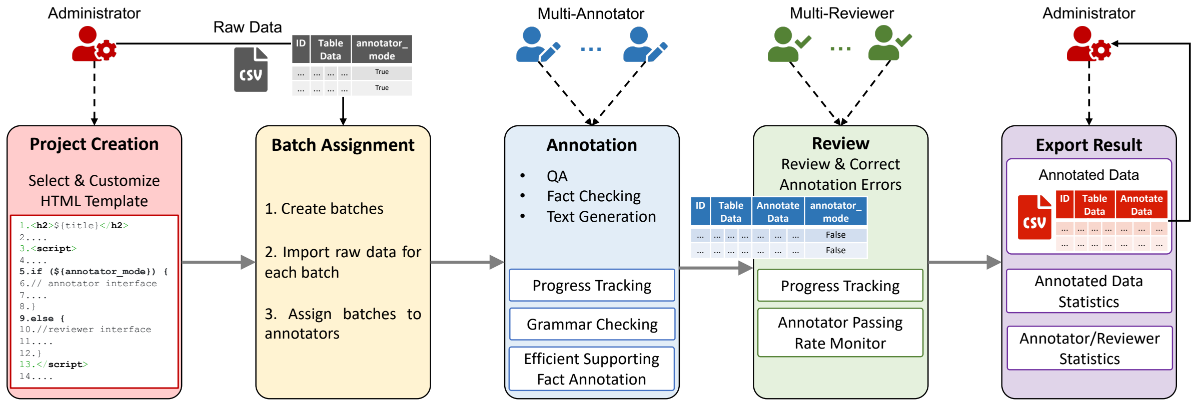


Figure 2: The overall workflow of TARAT.

Figure 3: The four design principles of TARAT: *quick deployment*, *better quality control*, *high productivity*, and *free accessibility*. Each principle comes with a series of feature designs that can make data annotation for table reasoning tasks more efficient and reliable.

to develop more powerful and effective table pre-training methods that can be applied to and achieve competitive performance on various types of table reasoning tasks.

5 TARAT Annotation Tools

In order to facilitate the construction of new table reasoning datasets for other researchers, we developed TARAT, the first open-source table reasoning annotation tool that supports the collaborative construction of various dataset types (i.e., question answering, fact checking, text generation). TARAT was designed, developed, and tested with the four design principles shown in Figure 3. As depicted in Figure 2, a typical annotation process using TARAT consists of the following five steps:

5.1 Annotation Project Creation

The administrator begins by accessing the *admin interface* of TARAT (Figure 4 in Appendix) to specify and set up an annotation project. Specifically, they need to select one of the annotation task templates provided by us as a starting point. These templates are customizable, so the administrator is allowed to adjust elements (e.g., annotator input type, display style of tabular data) to finalize a tailored annotation task specification.

5.2 Annotation Batch Assignment

The administrator can create multiple batches for an annotation project, with each batch containing multiple annotation tasks (i.e., we count annotating an example as one task). The division of the annotation project into multiple batches helps the administrators better organize and monitor the annotation progress. To initialize each batch, the administrators need to prepare raw annotation data in a csv file, with each line corresponding to an annotation task (Figure 5 in Appendix). Then the administrator can assign each batch to a specific group of annotators (Figure 6 in Appendix).

5.3 Annotation

Once the annotation batches are assigned, the annotators can begin working. In our preliminary study, we found that annotators and reviewers would spend a significant amount of time on typo/grammar correction and table evidence annotation (i.e, write down the row and column indices of relevant table cells). To improve annotation efficiency and quality, we accordingly implemented the following two features:

Grammar Checking We integrated the Grammarly Text Editor Plugin[2] into the TARAT annotation interface to help annotators detect and eliminate grammar and spelling mistakes. The annotators can view the editing suggestions by clicking the underlined text. They can then apply the sug-

[2] https://developer.grammarly.com/docs/

gested change by clicking “Accept”, or ignore it by clicking “Dismiss” (Figure 9 in Appendix).

Efficient Supporting Fact Annotation Previous work (Chen et al., 2020a, 2021) required annotators to manually write down the column and row indices of all relevant table cells (i.e., supporting fact), which is time-consuming and might introduce typos. To enable a more efficient supporting fact annotation, we implemented *cell highlight*, which allows the annotators to select (i.e., highlight) multiple relevant cells on the table as supporting facts (Figure 10 in Appendix). The indices of highlighted cells will be automatically recorded.

5.4 Annotation Review

Once an annotation batch is finished, the administrator can convert it to a reviewing batch at the TARAT *admin interface*, and assign the reviewing batch to a group of reviewers. The reviewers are expected to correct examples with annotation errors. The system will update the passing rate of each annotator, which the administrator can use to identify unqualified annotators and filter them out.

5.5 Annotation Result Export

After the review process, the annotated data can be exported by the administrator to a result file in CSV format (Figure 8 in Appendix). The administrator is also able to output the annotation statistics (e.g., passing rate, spent time on each example) for each annotator or reviewer, which can be used to determine annotation payment.

6 Related Work

Reasoning over Tabular Data The tasks related to reasoning over tables involves question answering (Pasupat and Liang, 2015; Zhong et al., 2017; Iyyer et al., 2017; Zhao et al., 2022a), fact checking (Chen et al., 2020c; Gupta et al., 2020), and faithful text generation (Chen et al., 2020b; Parikh et al., 2020; Zhao et al., 2023b) based on the information contained in the tables. Previous work mainly investigated how to develop a task-specific model that can work on one or two table reasoning datasets. More recently, inspired by the huge success of pre-trained language models (Devlin et al., 2019; Raffel et al., 2020), researchers have attempted to adopt the "pre-training, then fine-tuning" paradigm to develop models that can handle different kinds of table reasoning tasks with high performance (Herzig et al., 2020; Liu et al., 2022b; Jiang et al., 2022; Yang et al., 2022; Xie et al., 2022; Liu et al., 2022a). However, existing table pre-training methods have been evaluated on different datasets with varying configurations and developed as individual systems, resulting in difficulties in re-implementing them for performance comparison in future studies. The development of open-source libraries such as *Transformers* (Wolf et al., 2020) alleviate these issues to some extent, but they only cover a narrow range of table pre-training models and datasets. OPENRT implements existing table pre-training models in a unified and highly modularized framework, and provides standardized and comprehensive evaluation benchmarks for performance comparison.

Annotation Tools for Table Reasoning Tasks Existing annotation tools usually focus on the annotation with only textual input (Nakayama et al., 2018; Perry, 2021; Lin et al., 2022; Friedrich et al., 2022; Pei et al., 2022; Stodden and Kallmeyer, 2022). The development of table-relevant annotation tools is more complex as it requires the system to handle annotations on both textual and tabular input in a user-friendly manner. The current open-source table reasoning annotation tool, TABPERT (Jain et al., 2021), allows a user to update the table contents and associated hypotheses to generate counterfactual NLI examples. Compared to TABPERT, TARAT supports more types of table reasoning tasks, and can be hosted on a centralized server for large-scale distribution with a multi-person collaborative process. Furthermore, each component of TARAT is highly modularized and can be customized to meet the individual needs.

7 Conclusion

This work presents OPENRT, the first open-source framework for reasoning over tabular data, to reproduce existing table pre-training models for a standardized and fair performance comparison. OPENRT also enables users to quickly deploy their own models and datasets. Moreover, we developed TARAT to facilitate the construction of new table reasoning datasets by other researchers.

In the future, we will continue to add more table reasoning datasets and the latest released table pre-training models to OPENRT as part of regular updates. We welcome researchers and engineers to join us in developing, maintaining, and improving OPENRT and TARAT, in order to push forward the development of research on table reasoning.

Acknowledgements

We would like to dedicate this paper to the memory of Dr. Dragomir Radev. Dr. Radev's leadership, guidance, and expertise were instrumental in shaping the direction and quality of this project. His loss is deeply felt by all of us involved. We extend our heartfelt gratitude to Dr. Radev for his passion and dedication to the whole NLP community.

Ethical Consideration

The datasets included in OPENRT all use licenses that permit us to compile, modify, and publish the original datasets. TARAT is developed based on Turkle[3], which is released under the BSD-2-Clause license[4]. Both OPENRT and TARAT are also publically avaliable with the license BSD-2-Clause, which allows users to modify and redistribute the source code while retaining the original copyright.

References

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. *arXiv preprint arXiv:2004.10404*.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020b. 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. 2020c. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

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

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. 2022. AnnIE: An annotation platform for constructing complete open information extraction benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 44–60, Dublin, Ireland. Association for Computational Linguistics.

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

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.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Nupur Jain, Vivek Gupta, Anshul Rai, and Gaurav Kumar. 2021. TabPert : An effective platform for tabular perturbation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 350–360, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

[3]`https://github.com/hltcoe/turkle`

[4]`https://opensource.org/license/bsd-2-clause/`

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yupian Lin, Tong Ruan, Ming Liang, Tingting Cai, Wen Du, and Yi Wang. 2022. DoTAT: A domain-oriented text annotation tool. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022a. PLOG: Table-to-logic pretraining for logical table-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5531–5546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

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

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

Suixin Ou and Yongmei Liu. 2022. Learning to generate programs for table fact verification via structure-aware semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7624–7638, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. 2022. POTATO: The portable text annotation tool. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337, Abu Dhabi, UAE. Association for Computational Linguistics.

Tal Perry. 2021. LightTag: Text annotation platform. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 20–27, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Regina Stodden and Laura Kallmeyer. 2022. TS-ANNO: An annotation tool to build, annotate and evaluate text simplification corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 145–155, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng

Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022a. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022b. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9006–9018, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Lorenzo Jaime Flores, and Dragomir Radev. 2023a. Loft: Enhancing faithfulness and diversity for table-to-text generation via logic form control. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Xiangru Tang, Yumo Xu, Arman Cohan, and Dragomir Radev. 2023b. Qtsumm: A new benchmark for query-focused table summarization.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Fan Zhou, Mengkang Hu, Haoyu Dong, Zhoujun Cheng, Fan Cheng, Shi Han, and Dongmei Zhang. 2022. TaCube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2291, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

OpenRT includes following evaluation metrics for performance evaluation and comparison:

- **Accuracy** is scored as the number of correct predictions divided by total number of predictions.
- **BLEU** (Papineni et al., 2002) uses a precision-based approach, measuring the n-gram matches between the generated and reference statements.
- **ROUGE** (Lin, 2004) uses a recall-based approach, and measures the percentage of overlapping words and phrases between the generated output and reference one.
- **NLI-Acc** (Chen et al., 2020b) applies a natural language inference (NLI) model fine-tuned on TabFact (Chen et al., 2020c) to predict whether the generated sentence is entailed by source table.
- **SP-Acc** (Chen et al., 2020b) extracts the meaning representations from the generated sentence and executes them against the source table to verify the logical fidelity of the generated text.
- **BERTScore** (Zhang et al., 2020) computes the similarity between the generated sentence and reference ones using contextual word embeddings from BERT. For LogicNLG, which has multiple references for a source table, we compute the score by measuring the candidate with each reference and returning the highest score.
- **PARENT** (Dhingra et al., 2019) aligns n-grams from the reference and generated statements to the tabular data before computing their precision and recall. It achieves higher correlation with human judgement.

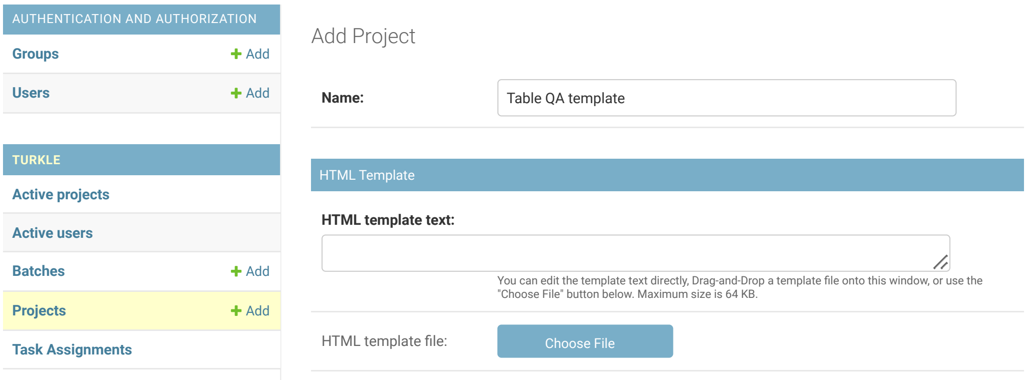
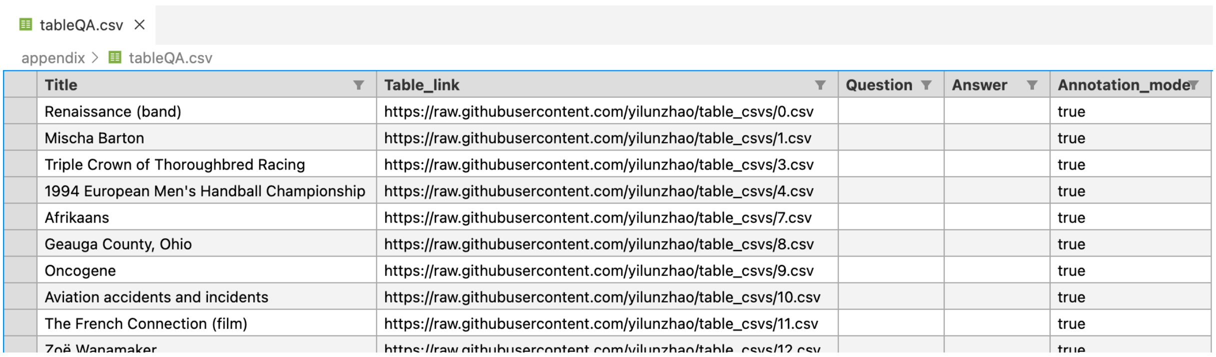


Figure 4: "Project Creation" in the administrator interface of TARAT. To set up a new annotation project, the administrator needs to choose, modify, and upload the HTML template for initializing the annotation interface.


tableQA.csv

appendix > tableQA.csv

Title	Table_link	Question	Answer	Annotation_mode
Renaissance (band)	https://raw.githubusercontent.com/yilunzhao/table_csvs/0.csv			true
Mischa Barton	https://raw.githubusercontent.com/yilunzhao/table_csvs/1.csv			true
Triple Crown of Thoroughbred Racing	https://raw.githubusercontent.com/yilunzhao/table_csvs/3.csv			true
1994 European Men's Handball Championship	https://raw.githubusercontent.com/yilunzhao/table_csvs/4.csv			true
Afrikaans	https://raw.githubusercontent.com/yilunzhao/table_csvs/7.csv			true
Geauga County, Ohio	https://raw.githubusercontent.com/yilunzhao/table_csvs/8.csv			true
Oncogene	https://raw.githubusercontent.com/yilunzhao/table_csvs/9.csv			true
Aviation accidents and incidents	https://raw.githubusercontent.com/yilunzhao/table_csvs/10.csv			true
The French Connection (film)	https://raw.githubusercontent.com/yilunzhao/table_csvs/11.csv			true
Zoë Wanamaker	https://raw.githubusercontent.com/yilunzhao/table_csvs/12.csv			true

Figure 5: An example of raw data stored in the csv file.

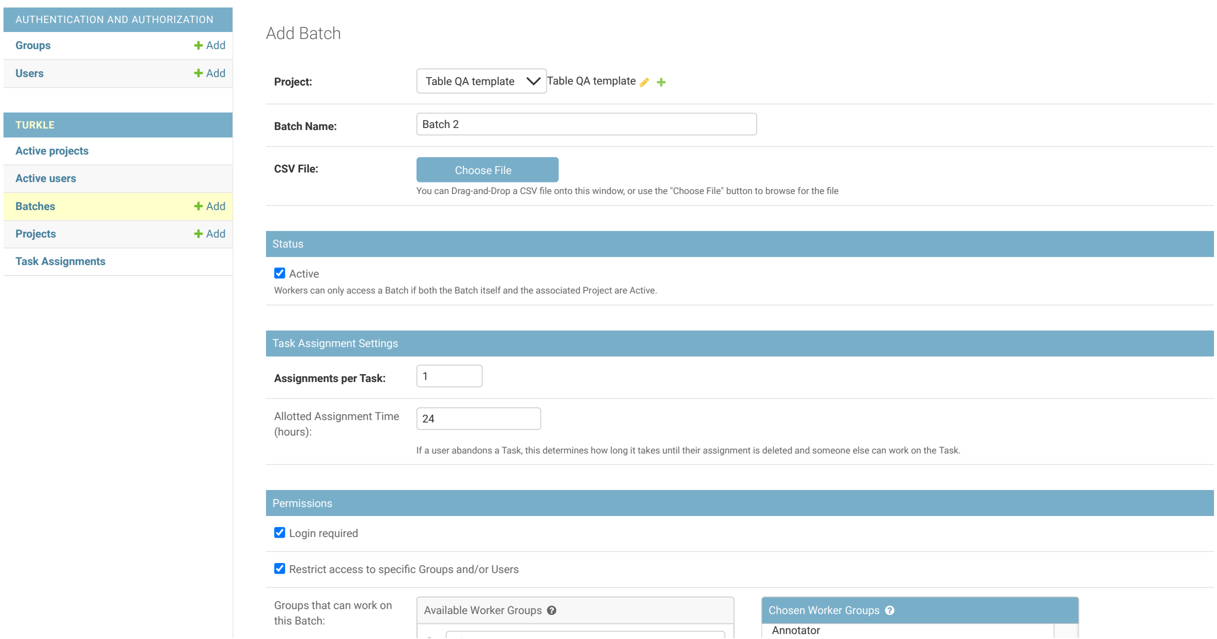


Figure 6: "Annotation Batch Creation" in the administrator interface of TARAT. The administrator can create an annotation batch by importing the raw data stored in a csv file, and assign the batch to a specific group of annotators.

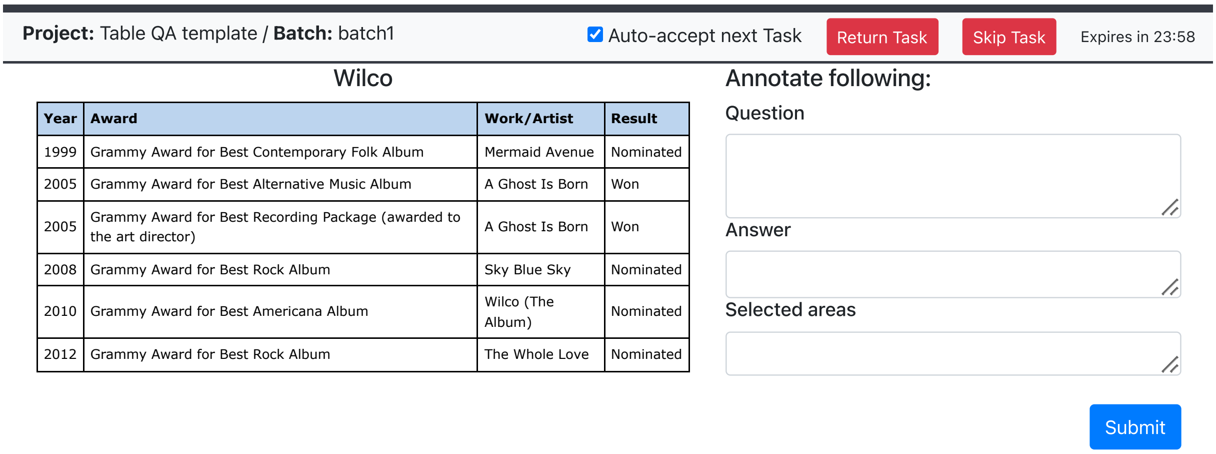


Figure 7: The annotation interface for Table QA task using provided HTML template.

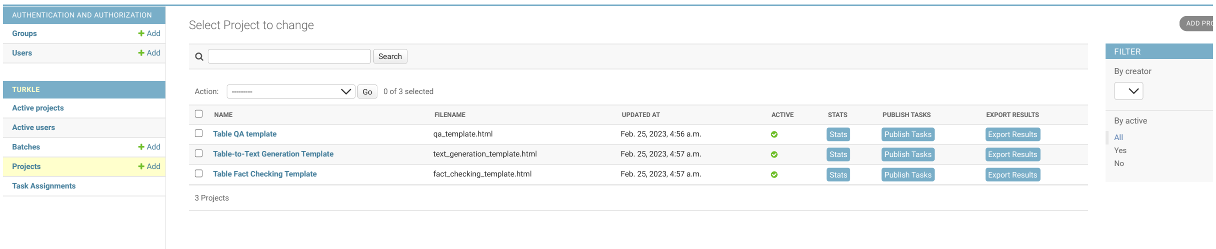


Figure 8: "Annotation Result Export" in the administrator interface of TARAT. The administrator can output the annotated data as well as the annotation statistics in CSV formats.

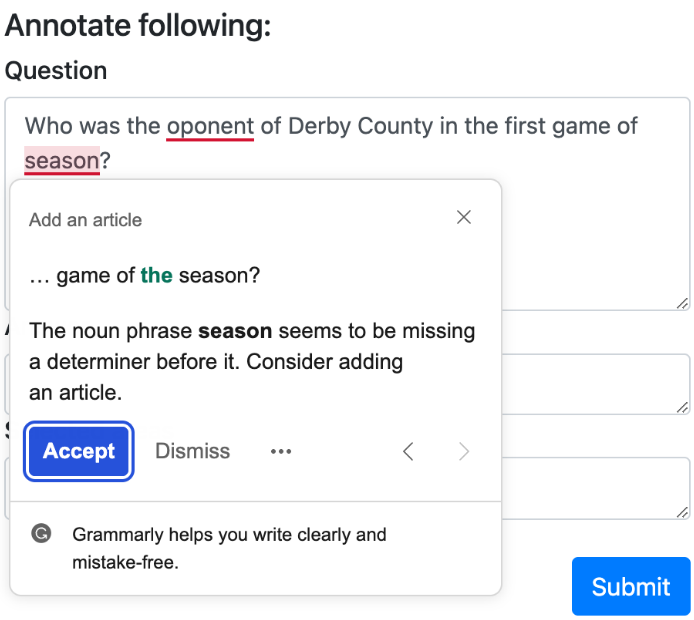
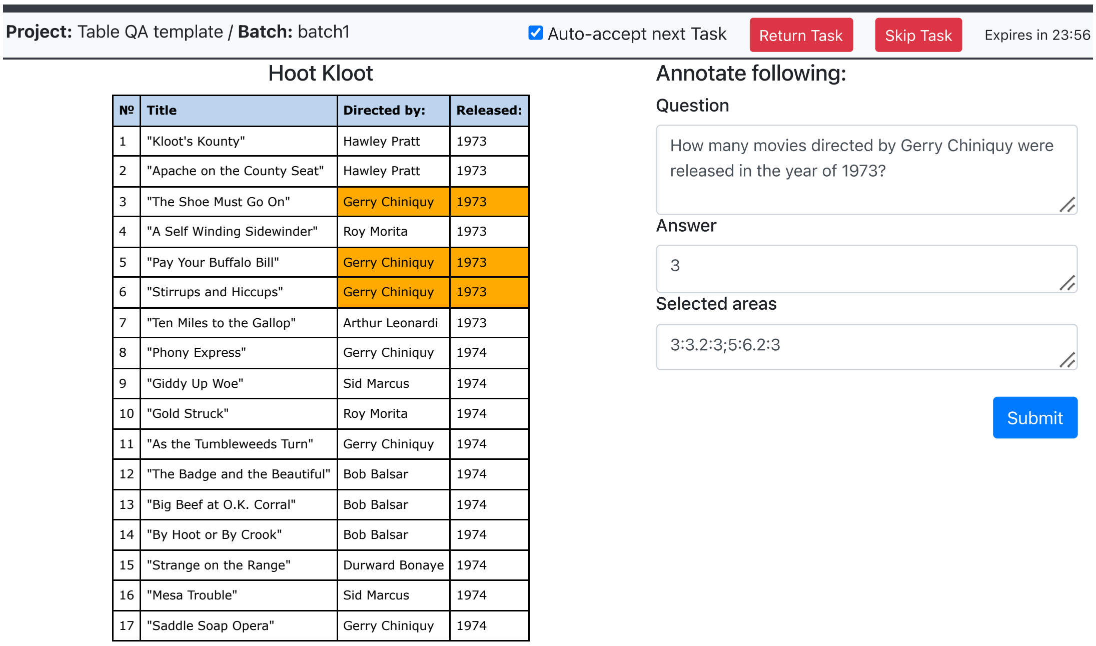


Figure 9: An example of grammar checking in TARAT. The annotation interface automatically detects the spelling errors and shows the editing suggestions to the annotator.



№	Title	Directed by:	Released:
1	"Kloot's Kounty"	Hawley Pratt	1973
2	"Apache on the County Seat"	Hawley Pratt	1973
3	"The Shoe Must Go On"	Gerry Chiniquy	1973
4	"A Self Winding Sidewinder"	Roy Morita	1973
5	"Pay Your Buffalo Bill"	Gerry Chiniquy	1973
6	"Stirrups and Hiccups"	Gerry Chiniquy	1973
7	"Ten Miles to the Gallop"	Arthur Leonardi	1973
8	"Phony Express"	Gerry Chiniquy	1974
9	"Giddy Up Woe"	Sid Marcus	1974
10	"Gold Struck"	Roy Morita	1974
11	"As the Tumbleweeds Turn"	Gerry Chiniquy	1974
12	"The Badge and the Beautiful"	Bob Balsar	1974
13	"Big Beef at O.K. Corral"	Bob Balsar	1974
14	"By Hoot or By Crook"	Bob Balsar	1974
15	"Strange on the Range"	Durward Bonaye	1974
16	"Mesa Trouble"	Sid Marcus	1974
17	"Saddle Soap Opera"	Gerry Chiniquy	1974

Figure 10: An example of *cell highlight* in TARAT. To annotate supporting facts, the annotators can directly select (i.e. highlight) the relevant table cells on the table. The indices of highlighted cells will be automatically recorded.