Adversarial Attacks on Tables with Entity Swap

Aneta Koleva[1,2], Martin Ringsquandl[1] and Volker Tresp[2]

[1]Siemens AG
[2]Ludwig Maximilian University of Munich

Abstract
The capabilities of large language models (LLMs) have been successfully applied in the context of table representation learning. The recently proposed tabular language models (TaLMs) have reported state-of-the-art results across various tasks for table interpretation. However, a closer look into the datasets commonly used for evaluation reveals an entity leakage from the train set into the test set. Motivated by this observation, we explore adversarial attacks that represent a more realistic inference setup. Adversarial attacks on text have been shown to greatly affect the performance of LLMs, but currently, there are no attacks targeting TaLMs. In this paper, we propose an evasive entity-swap attack for the column type annotation (CTA) task. Our CTA attack is the first black-box attack on tables, where we employ a similarity-based sampling strategy to generate adversarial examples. The experimental results show that the proposed attack generates up to a 70% drop in performance.

Keywords
Column Type Annotation, Adversarial Attack, Table Representation Learning

1. Introduction

Following the advancements of large language models (LLMs) in NLP, tabular language models (TaLMs) have emerged as state-of-the-art approaches to solve table interpretation (TI) tasks, such as table-to-class annotation [1], entity linking [2], and column type annotation (CTA) [3, 4, 5]. Similar to other deep neural networks, LLMs are sensitive to small input perturbation, which as adversarial examples can further be optimized to be imperceptible to humans [6]. Several works have studied adversarial attacks on LLMs, and it is becoming an increasingly important topic as LLMs are vastly being integrated into applications [7]. For the table modality, so far, sensitivity to perturbations has not been investigated in TaLMs for TI tasks. Hence, it is unclear which perturbation operations should be considered when attacking tables and how to make them imperceptible.

Using CTA as an example task, we phrase the novel problem of generating adversarial examples in entity tables. The existing models already report very high F1 scores on this task, and it is hard to judge by the performance of the model, how well it can generalize to unseen novel entities. In this direction, we design an evasive **entity-swap** attack that is motivated by a problem we observed. Namely, in two datasets commonly used for evaluation of the CTA task, WikiTables [3] and VizNet [4], there is a data leakage from entities from the training set into the test set.

In Table 1 we show the percentage of overlapped entities between the train and test set in the WikiTables dataset for the top 5 classes. The last 15 types in this dataset have 100 overlap among entities.

type	total	overlap	%
people.person	47852	29215	61.0
location.location	34073	21327	62.6
sports.pro_athlete	17588	10948	62.2
organization.organization	9904	7122	71.9
sports.sports_team	8207	6640	80.9

Table 1
Overlap of entities per type in the WikiTables dataset.

In Figure 1, both of the first two tables have a column named *Player* which contains the exact same set of entities, and is annotated with the same semantic types *Athlete* and *Person*. The third table shows an example of an adversarial table with an entity swap. In this table, the entities of the column *Player* are swapped with new, unseen entities of the same semantic type.

In the evaluation, we gradually increase the percentage of entities that we swap in the targeted columns, ranging from 20 % up to 100 % percent of the number of entities in the column. For choosing the adversarial entities, we propose a similarity-based strategy and compare it to sampling at random. Our evaluation demonstrates that swapping entities with the most dissimilar entity of the same type results in a substantial drop in performance (6% drop when replacing 20% of the entities per column up to 70% drop when replacing all of the entities).

2. Related Work

With the growing popularity of LLMs, the concern over their vulnerability to adversarial attacks also increased.

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — TaDA'23: Tabular Data Analysis Workshop, August 28 - September 1, 2023, Vancouver, Canada
*Corresponding author.

Table in training set

Tournament	Titles	Player(s)
Australian Open	10	Novak Djokovic
French Open	14	Rafael Nadal
Wimbledon	8	Roger Federer

Event — Athlete, Person

Table in test set

Player	Total	Years
Rafael Nadal	22	2005–2022
Novak Djokovic	22	2008–2023
Roger Federer	20	2003–2018

Athlete, Person

Adversarial table in test set

Player	Total	Years
Nikola Jokić	22	2005–2022
Maria Sharapova	22	2008–2023
Kobe Bryant	20	2003–2018

Athlete, Person

Figure 1: Entity-level adversarial example for table attack

A survey by Zhang et al. [8] presents a comprehensive overview of attacks against text, highlighting the challenges that arise when attacking discrete data such as text, compared to continuous data such as images. BERT-Attack [6] proposes an adversarial attack against the BERT model [9] using the model itself to generate the adversarial samples. A recent gradient-based text attack [10] presents a white-box attack which uses a parameterized adversarial distribution for sampling the adversarial samples. However, despite the popularity of adversarial attacks on text, the field of tabular data remains unexplored for potential vulnerabilities to such attacks.

The few works that have been proposed so far, [11, 12, 13], focus on white-box attacks and target traditional machine learning models trained on tabular data. The main goal of these attacks is when generating adversarial examples to preserve the distributional consistency of the features of the data. In these works, the datasets used for evaluation usually contain many numerical values, such as financial data or healthcare analytics data.

The goal of our work is to define table attacks against TaLMs which are used for TI tasks. To the best of our knowledge, we present the first work on adversarial attacks targeting these models. Our research differentiates from the prior work w.r.t (1) the model observed, (2) the technique employed for generating adversarial samples, and (3) the evaluation task.

3. CTA Adversarial Attack

We define a table as a tuple $T = (E, H)$, where $E = \{e_{1,1}, e_{1,2}, \dots, e_{i,j}, \dots, e_{n,m}\}$ is the set of table body entities for n rows and m columns. The table header $H = \{h_1, h_2, \dots, h_m\}$ is the set of corresponding m column header cells. We use $T_{[i,:]}$ to refer to the i-th row, e.g., $H = T_{[0,:]}$ and $T_{[:,j]} = \{h_j, e_{1,j}, \dots, e_{n,j}\}$ to refer to the j-th column of T.

CTA Model Let $\mathcal{T}$ be the input space of tables and let J be the space of all possible column indices, i.e., $J \subseteq \mathbb{N}$. Let $\mathcal{C}$ be the output space, denoting the set of semantic types. A CTA model is a multilabel classification function $h : \mathcal{T} \times J \rightarrow P(\mathcal{C})$, i.e., given a table $T \in \mathcal{T}$ and a column index $j \in J$ the CTA task is to assign a subset of classes from the power set of $\mathcal{C}$ to the corresponding column $T_{[:,j]}$.

CTA Attack Given classification model h, the goal of a CTA attack is to transform a (correctly classified) test input $(T, j) \in \mathcal{T} \times J$ into an (untargeted) adversarial sample (T', j) such that $h(T, j) \bigcap h(T', j) = \emptyset$. In addition to fooling the classification model, the transformation from T to T' should also be imperceptible for a human observer. In the CTA setting we define the imperceptibly condition to be met if all entities in column $T'_{[:,j]}$ are of the same class as the unmodified column. Formally, $\forall e' \in T'_{[:,j]} \forall e \in T_{[:,j]} : c(e') = c(e)$, where $c \in \mathcal{C}$ represents the most specific class assigned to the column $T_{[:,j]}$.

3.1. Entity Swap Attack

In principle, a CTA attack can apply transformations to the full table T; however, most importantly, it should focus on $T_{[:,j]}$. Our attack, called **entity-swap**, follows a two-step approach inspired by adversarial attacks on LLMs [6, 14]. First, it picks a set of key entities $\{e_i \in T_{[:,j]}\}$. The number of key entities can be controlled as a percentage p of the entities in the original column. In a second step, every key entity e_i is swapped with an adversarial entity $e_i = e_i'$ that most likely changes the predicted class from the ground truth. The proposed attack is a black-box attack, meaning we only have access to the predictions scores of the classifier.

3.2. Key Entities

Finding which are the key entities to swap can increase the rate of success of the attack. In the case of the CTA task, the most informative entities are those which, when replaced, the model will misclassify the column. To find those entities, we calculate an importance score for every entity in the attacked column.

The output from the classification model h for a column $T_{[:,j]}$ is the logit vector $\mathbf{o_h}(T, j) \in \mathbb{R}^k$, where k is the number of ground-truth classes assigned to (T, j). We calculate the importance score for entity $e_i \in T_{[:,j]}$ as the

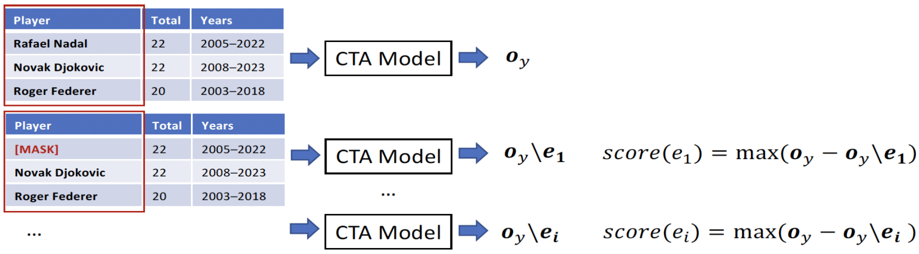

Figure 2: Calculation of importance scores.

difference between the logit output of the model for the ground truth classes when the entity is *in* the column, denoted as $\mathbf{o_h}$, and the logit output of the model when the entity is *replaced* with the [MASK] token, denoted as $\mathbf{o_h} \backslash \mathbf{e_i}$. Since, the CTA task is evaluated under the multi-label setting, we always take the maximum importance score for an entity.

$$score(e_i) = max(\mathbf{o_h} - \mathbf{o_h} \backslash \mathbf{e_i}) \tag{1}$$

Figure 2 shows an example of how the importance score is calculated. We calculate $\mathbf{o_h}$ as the logit output of the model without any perturbation, while $\mathbf{o_h}/\mathbf{e_1}$ represents the logit output of the model when the entity *Rafael Nadal* is masked. After calculating the importance score for every entity in the column, we select the top p percent of entities ($p \in 20, 40, 60, 80, 100$) based on their importance scores and substitute them with adversarial entities. By sorting the entities according to their importance scores, we ensure that the attack consistently targets the key entities within the targeted column.

3.3. Adversarial Entities

After identifying the key entities, the next step involves sampling adversarial entities for swapping. In order to adhere to the perceptibility assumption, we constrain the search space to include only entities belonging to the same class as the attacked column. Subsequently, we use a similarity-based strategy to sample adversarial entities.

Let $e_i \in T_{[:,j]}$ be the key entity from the attacked column, and let $c \in \mathscr{C}$ be the most specific class of this column. We use an embedding model to generate a contextualized representation for both the original entity, $\mathbf{e_i}$, and all entities of the same class $A_c = \{\mathbf{e'_1}, \mathbf{e'_2}, \ldots, \mathbf{e'_k}\}$, such that $c(e_i) = c(e'_k)$ where $e'_k \in A_c$. Next, we calculate the cosine similarity between the original entity and each entity from the set A_c. As an adversarial example, we take the most dissimilar entity from the original entity, such that $e'_i = \text{argmax}_{e'_k} \textsc{CosineSimilarity}(\mathbf{e_i}, \mathbf{e'_k})$. We then swap the original entity e_i with the adversarial entity e'_i.

As we describe in the introduction, there is a substantial overlap of entities between the train and test set. Therefore, we propose two different sampling sets for adversarial entities. The first, is the set of entities per class from the WikiTables test dataset [3]; we refer to this set as *test set*. The second set contains only novel entities, i.e., entities that also appear in the training set, are removed from the test set. We refer to this set as the *filtered set*.

Metadata Attack In addition to the proposed attack method for column values, we also introduce an attack specifically targeting column headers, considering that they often indicate the class of a column. However, in this case, we use an independent embedding model to identify similar entities instead of swapping with column names from the same class. For the generation of adversarial samples in the column headers, we first generate embeddings for the original column names and then substitute the column names with their synonyms. The library *TextAttack* [14] was used to generate the embeddings, and based on the embeddings to retrieve the synonyms for the column names.

4. Evaluation

Model We evaluate the performance of the CTA attack on the TURL model [3], which has been fine-tuned for the CTA task and uses only entity mentions. We use the WikiTables dataset for evaluation. We follow their evaluation procedure and report the achieved F1 score, precision, and recall.

To evaluate the influence of the proposed strategy for sampling adversarial samples, we compare it to a random sampling of adversarial entities. Similarly, to evaluate the influence of the importance scores, we compare with random sampling when choosing which entities to swap.

4.1. Results

Table 2 shows the results of the CTA attack when swapping entities by their importance scores and sampling

% perturb.	F1	P	R
0 (original)	88.86	90.54	87.23
20	83.4 (6%)	90.3 (0.2%)	77.8 (11%)
40	72.0 (19%)	87.9 (3%)	60.9 (30%)
60	55.3 (38%)	80.4 (11%)	42.1 (52%)
80	39.9 (55%)	67.7 (25%)	28.4 (67%)
100	26.5 (70%)	50.8 (44%)	17.9 (80%)

Table 2
Adversarial attack on the entities. The adversarial entities are sampled by their semantic similarity from the original entity.

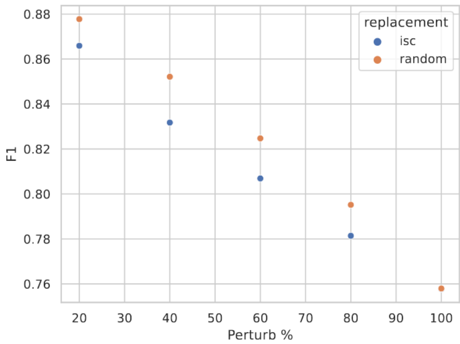

Figure 3: Adversarial samples from the test set, replacing entities at random vs using the importance scores.

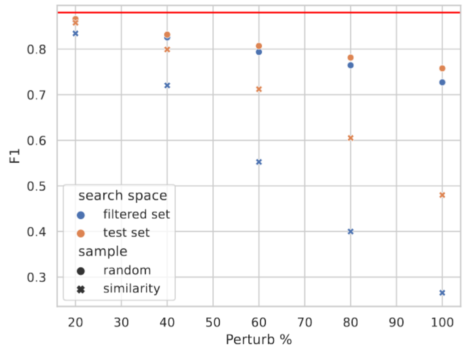

Figure 4: Sampling adversarial entities from the test set vs the filtered set, at random and using the similarity strategy.

%	F1	P	R
0 (original)	90.24	89.91	90.58
20	78.4 (13%)	81.1 (10%)	76.0 (16%)
40	77.1 (15%)	80.7 (10%)	73.8 (19%)
60	75.2 (17%)	79.1 (12%)	72.2 (20%)
80	65.1 (28%)	71.4 (22%)	60.4 (33%)
100	51.2 (43%)	60.4 (33%)	44.4 (51%)

Table 3
Attack on the column names where the adversarial samples are their synonyms.

adversarial entities using the similarity-based strategy from the filtered set. We notice that as we increase the percentage of swapped entities, the performance of the model drops, even though the perturbed entities are of the same semantic type as the original entities. Another observation is that the drop in the F1 score is attributed to the sharp decline of the recall.

Effect of the importance score Figure 3 shows the benefit of using the importance scores. We notice that the drop in F1 score is around 3% higher when using the importance scores. This is consistent, regardless if we are substituting 20% or 80% of the entities, which suggests that the importance scores consistently identify entities that have a greater influence on the model's performance.

Effect of the sampling strategy Figure 4 shows the difference in F1 score drop when sampling adversarial entities from the test set versus the filtered set. The original F1 score is represented by the red line. Additionally, here we illustrate the advantages of using the similarity-based strategy over a random-based sampling of adversarial examples. For both cases, when sampling adversarial entities from the test and filtered set, the similarity-based strategy for sampling induces sharper drop of the F1 score. This suggests that this approach is successful in selecting entities that are more likely to cause misclassifications or confusion for the classification model.

Effect of perturbing the table metadata To evaluate the relevance of the column header for the CTA task, we also propose an adversarial attack specific to the TURL model [3], which uses only the table metadata. Table 3 shows the effect of perturbing the table metadata. We observe similar results here, as we increase the percentage of perturbed column names, all the evaluation metrics decline. This indicates that the model's reliance on specific column names, affects its ability to accurately classify and predict the correct class.

5. Conclusion

In this paper, we introduce the formalization of an adversarial attack targeting TaLMs. Additionally, we identify and highlight an issue concerning the evaluation of the CTA task. The evaluation showed that TaLMs are susceptible to adversarial attacks. Even subtle modifications to the entities, guided by similarity, can lead to significant changes in the model's predictions and subsequently affect the F1 score. In future, we will extend our evaluation with more sophisticated attacks, targeting also other models used for table interpretation tasks.

References

[1] A. Koleva, M. Ringsquandl, M. Joblin, V. Tresp, Generating table vector representations, in: Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, online, October 25, 2021, volume 3034 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-3034/paper7.pdf.

[2] V. Huynh, Y. Chabot, T. Labbé, J. Liu, R. Troncy, From heuristics to language models: A journey through the universe of semantic table interpretation with DAGOBAH, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, SemTab 2021, co-located with the 21st International Semantic Web Conference, ISWC 2022, Virtual conference, October 23-27, 2022, volume 3320 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 45–58. URL: https://ceur-ws.org/Vol-3320/paper6.pdf.

[3] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, Turl: Table understanding through representation learning, Proc. VLDB Endow. 14 (2020) 307–319.

[4] M. Hulsebos, K. Z. Hu, M. A. Bakker, E. Zgraggen, A. Satyanarayan, T. Kraska, Ç. Demiralp, C. A. Hidalgo, Sherlock: A deep learning approach to semantic data type detection, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM, 2019, pp. 1500–1508. URL: https://doi.org/10.1145/3292500.3330993. doi:10.1145/3292500.3330993.

[5] Y. Suhara, J. Li, Y. Li, D. Zhang, Ç. Demiralp, C. Chen, W. Tan, Annotating columns with pre-trained language models, in: SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022, ACM, 2022, pp. 1493–1503. URL: https://doi.org/10.1145/3514221.3517906. doi:10.1145/3514221.3517906.

[6] L. Li, R. Ma, Q. Guo, X. Xue, X. Qiu, BERT-ATTACK: adversarial attack against BERT using BERT, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 6193–6202. URL: https://doi.org/10.18653/v1/2020.emnlp-main.500. doi:10.18653/v1/2020.emnlp-main.500.

[7] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, M. Fritz, Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection, 2023. arXiv:2302.12173.

[8] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: A survey, ACM Trans. Intell. Syst. Technol. 11 (2020) 24:1–24:41. URL: https://doi.org/10.1145/3374217. doi:10.1145/3374217.

[9] J. Devlin, M. Chang, K. Lee, K. Toutanova, 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[10] C. Guo, A. Sablayrolles, H. Jégou, D. Kiela, Gradient-based adversarial attacks against text transformers, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Association for Computational Linguistics, 2021, pp. 5747–5757. URL: https://doi.org/10.18653/v1/2021.emnlp-main.464. doi:10.18653/v1/2021.emnlp-main.464.

[11] V. Ballet, X. Renard, J. Aigrain, T. Laugel, P. Frossard, M. Detyniecki, Imperceptible adversarial attacks on tabular data, CoRR abs/1911.03274 (2019). URL: http://arxiv.org/abs/1911.03274. arXiv:1911.03274.

[12] F. Cartella, O. Anunciação, Y. Funabiki, D. Yamaguchi, T. Akishita, O. Elshocht, Adversarial attacks for tabular data: Application to fraud detection and imbalanced data, in: Proceedings of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) co-located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021, volume 2808 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2808/Paper_4.pdf.

[13] Y. Mathov, E. Levy, Z. Katzir, A. Shabtai, Y. Elovici, Not all datasets are born equal: On heterogeneous tabular data and adversarial examples, Knowl. Based Syst. 242 (2022) 108377. URL: https://doi.org/10.1016/j.knosys.2022.108377. doi:10.1016/j.knosys.2022.108377.

[14] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, Y. Qi, Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 119–126. URL: https://doi.org/10.18653/v1/2020.emnlp-demos.16. doi:10.18653/v1/2020.emnlp-demos.16.