Auto-Tables: Synthesizing Multi-Step Transformations to Relationalize Tables without Using Examples

Peng Li, Yeye He, Cong Yan, Yue Wang, Surajit Chaudhuri

Published: 01 July 2023

Published in PVLDB Volume 16, Issue 11

Presented by Evan Yuchen Zhu, Hangtian Zhu, Hanqi Hua, Minzhi Wang, Peter James Feng

What is Auto-Tables

- Automatically converts complex, non-relational tables into standard relational formats for easy querying, using predefined transformations without needing user input

- Key Features:
    - Set of predefined transformation operators
    - Computer-vision inspired model architecture
    - Automatic table relationalization
    - Efficient and Fast

Why Auto-Tables

- Sampled hundreds of user spreadsheets (in Excel) and web tables (from Wikipedia)
- Around **30-50%** tables do not conform to the relational standard
- Require complex manual table-restructuring transformations before these tables can be queried easily using SQL-based tools.
- Prevalent at a very large scale (**millions** of tables like these)

Why Auto-Tables E.g.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Product**	**Product Category**	**Store**	19-Oct	20-Oct	21-Oct	22-Oct	23-Oct	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct
2	Huffy 18 in. Boys Bike	Sports	s_sk_101	5	3	4	15	19	2	5	11	3	9
3	Kent 18 In. Boy's BMX Bike	Sports	s_sk_101	8	5	11	12	8	14	7	5	9	9
4	HP 11 in. Chromebook 16W64	Electronics	s_sk_102	17	9	14	5	19	18	17	18	10	7
5	Mainstays Computer Desk	Furniture	s_sk_103	6	4	1	16	8	7	6	9	8	20

	A	B	C	D	E
1	**Product**	**Product Ca**	**Store**	Date	Unit Sold
2	Huffy 18 in. Boys Bike	Sports	s_sk_101	19-Oct	5
3	Huffy 18 in. Boys Bike	Sports	s_sk_102	20-Oct	3
4	Huffy 18 in. Boys Bike	Sports	s_sk_103	21-Oct	4
5	Huffy 18 in. Boys Bike	Sports	s_sk_104	22-Oct	15
6	...	...	...	...	...
7	Kent 18 In. Boy's BMX Bike	Sports	s_sk_101	19-Oct	8
8	Kent 18 In. Boy's BMX Bike	Sports	s_sk_102	20-Oct	5

(a) Stack: transforming homogeneous columns into rows.
The colored columns in input are homogeneous and should collapse together.

	A	B	C	D	E	F	G	H	I	J	K
1	**Country**	**Region**	2018 - Revenue ($K)	2018 - Units Sold	2018 - Margin %	2019 - Revenue ($K)	2019 - Units Sold	2019 - Margin %	2020 - Revenue ($K)	2020 - Units Sold	2020 - Margin %
2	Albania	Europe	$10	224	4%	$12	269	4%	$16	350	4%
3	Australia	Asia Pacific	$2,492	54824	13%	$2,990	65789	14%	$3,888	85525	16%
4	Argentina	South America	$495	10890	9%	$594	13068	11%	$772	16988	13%
5	Belarus	Europe	$29	638	10%	$35	766	9%	$45	995	9%
6	Belgium	Europe	$384	8448	15%	$461	10138	15%	$599	13179	15%
7	Brazil	South America	$102	2244	12%	$122	2693	13%	$159	3501	15%
8	Canada	North America	$4,039	88858	15%	$4,847	106630	17%	$6,301	138618	18%

	A	B	C	D	E	F
1	**Country**	**Region**	**Year**	Revenue ($K)	Units Sold	Margin %
2	Albania	Europe	2018	$10	224	4%
3	Albania	Europe	2019	$12	269	4%
4	Albania	Europe	2020	$16	350	4%
5	Australia	Asia Pacific	2018	$2,492	54824	13%
6	Australia	Asia Pacific	2019	$2,990	65789	14%
7	Australia	Asia Pacific	2020	$3,888	85525	16%
8	Argentina	South Ame	2018	$495	10890	9%
9	Argentina	South Ame	2019	$594	13068	11%
10	Argentina	South Ame	2020	$772	16988	13%

(b) Wide-to-long: transforming repeating column groups into rows.
The colored col-groups in input have repeating patterns and should collapse.

Why Auto-Tables E.g.

	A	B	C	D	E
1		HOTEL MISION JURIQUILLA	RAMADA ENCORE HOTEL	FOUR POINTS BY SHERATON	PLAZA CAMELINAS HOTEL
2	Single Room	1030	920	1150	789
3	Lodging tax	2.50%	3.50%	3.50%	2.50%
4	I.V.A.	0.16	0.16	0.16	0.16
5	Address	Centro, cp. 76000	Juriquilla, C.P. 76230	Jurica, C.P. 76127	La Capilla, 76170
6	Phone	(442) 234-0000 ex. 547	(442) 690-9400	(4429 103-3030	(442) 192-3900
7	Webpage	http://www.htlmision.com	www.hotelesencore.com	http://www.starwood.com	www.plazacamelinas.com
8	Stars	*****	****	*****	*****

	A	B	C	D	E	F	G	H
1		Single Room	Lodging tax	I.V.A.	Address	Phone	Webpage	Stars
2	HOTEL MISION JURIQUILLA	1030	2.50%	0.16	Centro, cp. 76000	(442) 234-0000 ex 547	http://www.htlmision.com	*****
3	RAMADA ENCORE HOTEL QRO.	920	3.50%	0.16	Juriquilla, C.P. 76230	(442) 690-9400	www.hotelesencore.com	****
4	FOUR POINTS BY SHERATON	1150	3.50%	0.16	Jurica, C.P. 76127	(4429 103-3030	http://www.starwood.com	*****
5	PLAZA CAMELINAS HOTEL	789	2.50%	0.16	La Capilla, 76170	(442) 192-3900	www.plazacamelinas.com	*****

(c) Transpose: transforming rows to columns and vice versa.
The colored rows in input have homogeneous content in the horizontal direction.

	A
1	Found: 21-Oct-19 10:21:14
2	Title: Canon EF 100mm f/2.8L Macro IS USM
3	Price: 6900 kr
4	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=161065896
5	Found: 21-Oct-19 10:21:15
6	Title: Canon EF 85mm f/1.8 USM Medium
7	Price: 7500 kr
8	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=155541389
9	Found: 21-Oct-19 10:22:46
10	Title: Panasonic Lumix G 25mm F1.4 ASPH
11	Price: 3200 kr
12	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=161066674

	A	B	C	D
1	Found: 21-Oct-19 10:21:14	Title: Canon EF 100mm f/2.8L Macro IS USM	Price: 6900 kr	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=161065896
2	Found: 21-Oct-19 10:21:15	Title: Canon EF 85mm f/1.8 USM Medium	Price: 7500 kr	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=155541389
3	Found: 21-Oct-19 10:22:46	Title: Panasonic Lumix G 25mm F1.4 ASPH	Price: 3200 kr	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=161066674
4	Found: 21-Oct-19 10:24:50	Title: Panasonic Lumix DMC-G7 Mirrorless	Price: 6900 kr	Link: https://www.finn.no/bap/forsale/ad.html?finnkode=161827163

(d) Pivot: transforming repeating row groups into columns.
The colored rows in input have repeating patterns that should become cols.

Why Auto-Tables concl.

- Both technical and non-technical users complain about the difficulty of doing manual transformations
    - Many questions on Excel & Tableau forums and StackOverflow
- Auto-Tables:
    - Automatically synthesize pipelines with multi-step transformations
    - Over **70%** of success rate on test cases at interactive speeds
    - Without requiring any input from users
    - Effective tool for both technical and non-technical users to prepare data for analytics

Table Restructuring Operators

- Eight table restructuring operators cover most scenarios of relationalizing tables
- Need to predict exactly which operation + what parameter values
- Need a “None” operator to represent tables that don’t need transformation.

Table 1: Auto-Tables DSL: table-restructuring operators and their parameters to “relationalize” tables. These operators are common and exist in many different languages, like Python Pandas and R, sometimes under different names.

DSL operator	Python Pandas equivalent	Operator parameters	Description (example in parenthesis)
stack	melt [18]	start_idx, end_idx	collapse homogeneous cols into rows (Fig. 1a)
wide-to-long	wide_to_long [22]	start_idx, end_idx, delim	collapse repeating col-groups into rows (Fig. 1b)
transpose	transpose [21]	-	convert rows to columns and vice versa (Fig. 1c)
pivot	pivot [19]	repeat_frequency	pivot repeating row-groups into cols (Fig. 1d)
explode	explode [16]	column_idx, delim	convert composite cells into atomic values
ffill	ffill [17]	start_idx, end_idx	fill structurally empty cells in tables
subtitles	copy, ffill, del	column_idx, row_filter	convert table subtitles into a column
none	-	-	no-op, the input table is already relational

Problem Statement

Definition 1. Given an input table T, and a set of operators $\mathbf{O} = \{stack, transpose, pivot, \ldots\}$, where each operator $O \in \mathbf{O}$ has a parameter space $P(O)$. Synthesize a sequence of multi-step transformations $M = (O_1(p_1), O_2(p_2), \ldots, O_k(p_k))$, with $O_i \in \mathbf{O}$ and $p_i \in P(O_i)$ for all $i \in [k]$, such that applying each step $O_i(p_i) \in M$ successively on T produces a relationalized version of T.

- Generate a series of operators & parameters that relationalizes the table
- Parameter spaces can be large
  - Table with 50 columns can have 50x50=2500 combinations for *start_idx*, *end_idx*
  - This increases multiplicatively for multi-step transformations. 2500^2 = ~6M
  - Need to predict **exact** transformation and parameters. Cannot be off!
    - transpose(), stack("2015", "2020")

	B	C	D	E	F
1	Adams Elementary	Aki Kurose Middle School	Alki Elementary	B.F. Day Elementary	...
2	ES	MS	ES	ES	...
3	553	685	373	282	...
4	580	719	377	296	...
5	609	754	380	310	...
6	638	791	384	326	...
7	670	829	388	341	...
8	702	870	392	358	...

transpose

	A	B	C	D	E	F	G	H
1	School name	GradeID	2015	2016	2017	2018	2019	2020
2	Adams Elementary	ES	553	580	609	638	670	702
3	Aki Kurose Middle School	MS	685	719	754	791	829	870
4	Alki Elementary	ES	373	377	380	384	388	392
5	B.F. Day Elementary	ES	282	296	310	326	341	358
6	...	...	...	...	...	...	...	...

stack

	A	B	C	D
1	School name	GradeID	Year	Num
2	Adams Elementary	ES	2015	553
3	Adams Elementary	ES	2016	580
4	Adams Elementary	ES	2017	609
5	Adams Elementary	ES	2018	638
6	Adams Elementary	ES	2019	670
7	Adams Elementary	ES	2020	702
8	Aki Kurose Middle School	MS	2015	685
9	Aki Kurose Middle School	MS	2016	719
10	...	...	...	...

Architecture Overview

Offline

1. Training data generation using inverse operators
2. Input-only synthesis model training
3. Reranking model for outputs from step 2

Online

1. Generate outputs using input-only synthesis model
2. Use reranking model with outputs from step 1 to determine most likely final table

Architecture Overview

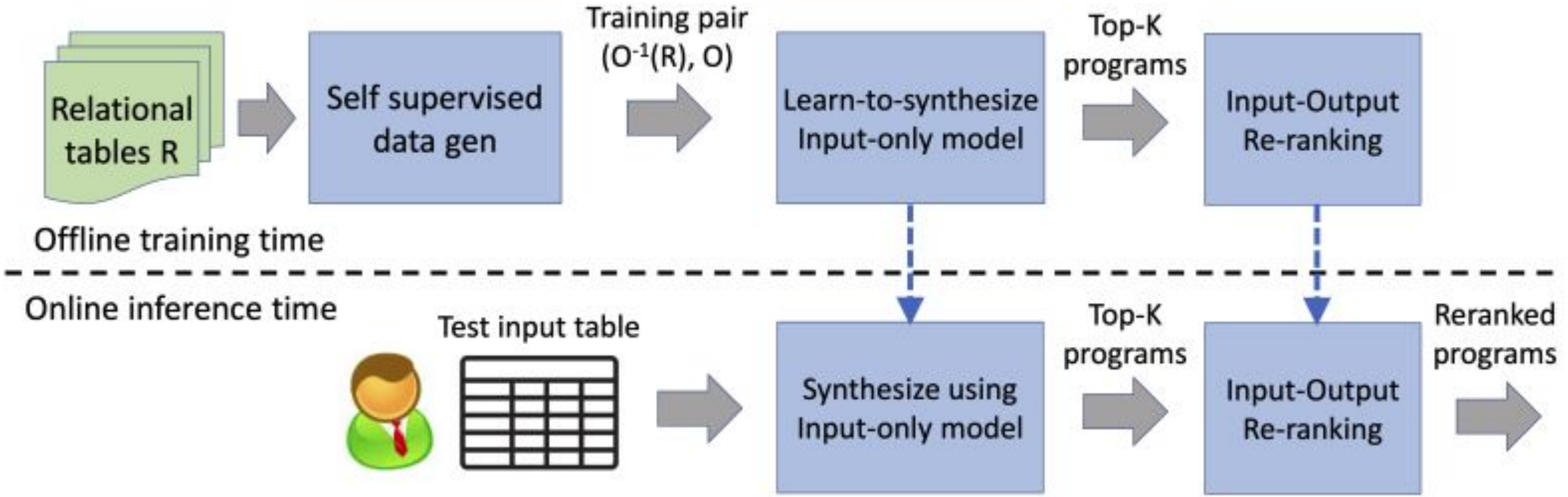


Figure 5: Architecture overview of AUTO-TABLES

Semi Supervised Training Data Generation

- Main challenge: not enough existing labeled training data for CV model
- Leverage **inverse operators** to generate high volume of training data
  - Inverse of “transpose” is “transpose”
  - Inverse of “wide-to-long” is [“stack”, “split”, “pivot”]
- Data augment from existing relational tables.
  - Cropping - randomly sample contiguous blocks of rows or columns
  - Shuffling - randomly reordering rows or columns
- 15k Relational Tables * 20 augmentations

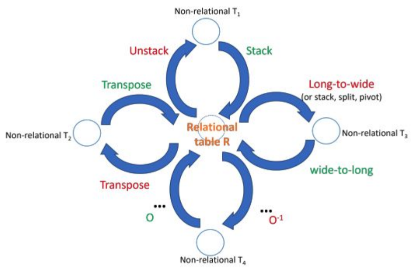


Figure 6: Leverage inverse operators to generate training data. In order to learn-to-synthesize operator O, we can start from any relational table R, apply its inverse operator O^{-1} to obtain $O^{-1}(R)$. Given $T = O^{-1}(R)$ as an input table, we know O must be its ground-truth transformation, because $O(O^{-1}(R)) = R$.

Semi Supervised Training Data Generation

Algorithm 1: Auto-gen training examples

---

input :DSL operators $\mathbf{O}$, large collections of relational tables $\mathbf{R}$

output :Training table-label pairs: (T, O_p)

$E \leftarrow \{\}$

foreach O *in* $\mathbf{O}$ **do**

    foreach R *in* $\mathbf{R}$ **do**

        foreach R' *in Augment(R)* `// Crop rows and columns`

        do

            $p \leftarrow$ sample valid parameter from space $P(O)$

            $O_{p'}^{-1} \leftarrow$ construct the inverse of O_p

            $T \leftarrow O_{p'}^{-1}(R')$

            $E \leftarrow E \cup \{(T, O_p)\}$

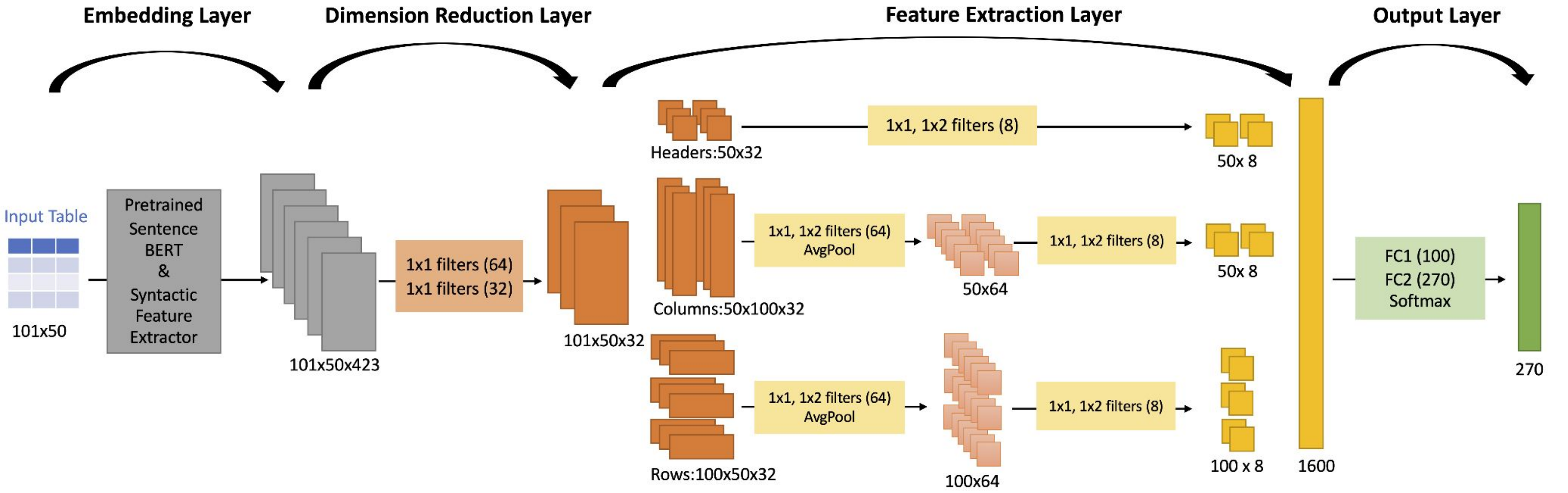
return all training examples E

Input-only Synthesis

After obtaining large amounts of training data in the form of (T, O_p) using self-supervision, we now describe our "input-only" model that takes T as input, to predict a suitable transformation O_{p}, and it has two parts:

1. **Model architecture**
2. **Training and inference**

Model architecture



Model architecture

1. Table embedding layers capture information about:

- "*syntactic feature*" (e.g., data-type,string-length, punctuation, etc.) using syntactic feature extractor
- "*semantic features*" (e.g., people-names, company-names, etc.) using pretrained sentence BERT

2. Dimension reduction layers:

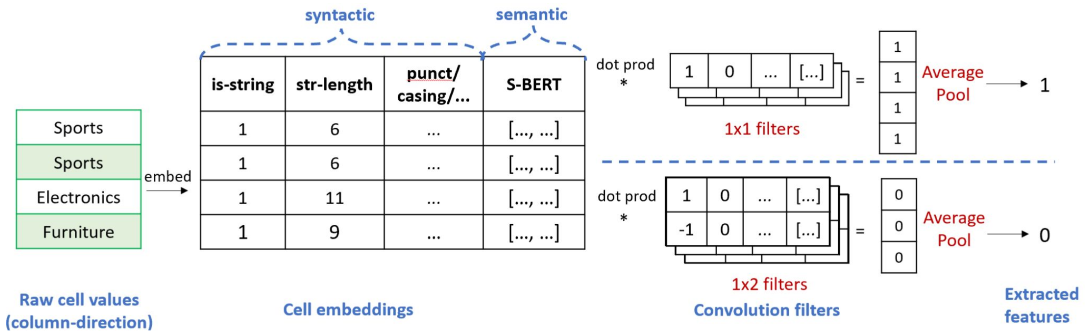
- Using two convolution layers with 1 ×1 kernels, to reduce the dimensionality from 423 to 64 and then to 32, to produce a $n \times m \times 32$ tensor.

Model architecture

3. Feature extraction layers:

- Using convolution filters similar to CNN with 1x2 and 1x1 convolution filters followed by average-pooling, in both row and column directions, to represent rows/columns/header.

Example:



4. Output layers: Use two fully connected layers followed by softmax classification to produce a 270 dimensions output vector that encodes both the predicted operator type, and its parameters for a given T.

Training and inference

Training time: Loss Function is the summation of cross-entropy loss

$$Loss(T) = L(O, \hat{O}) + \sum_{p_i \in P, \hat{p}_i \in \hat{P}} L(p_i, \hat{p_i})$$

$$L(y, \hat{y}) = -\sum_{i=1}^{n} y_i log(\hat{y}_i)$$

Inference time: Synthesizing transformations

$$Pr(O_P|T) = Pr(O) \cdot \prod_{p_i \in P} Pr(p_i)$$

$$O_P^* = \arg\max_{O,P} Pr(O_P|T)$$

Training and inference

- O^*_p gives us the most likely one-step transformation given T. And tables may require multiple transformation steps for our task.

- To synthesize multi-step transformations, one possible solution is consider only the top-1 choice at each step, but it’s not optimal.

- Therefore, we consider top-k choices at each step to find the most likely multi-step transformations overall.

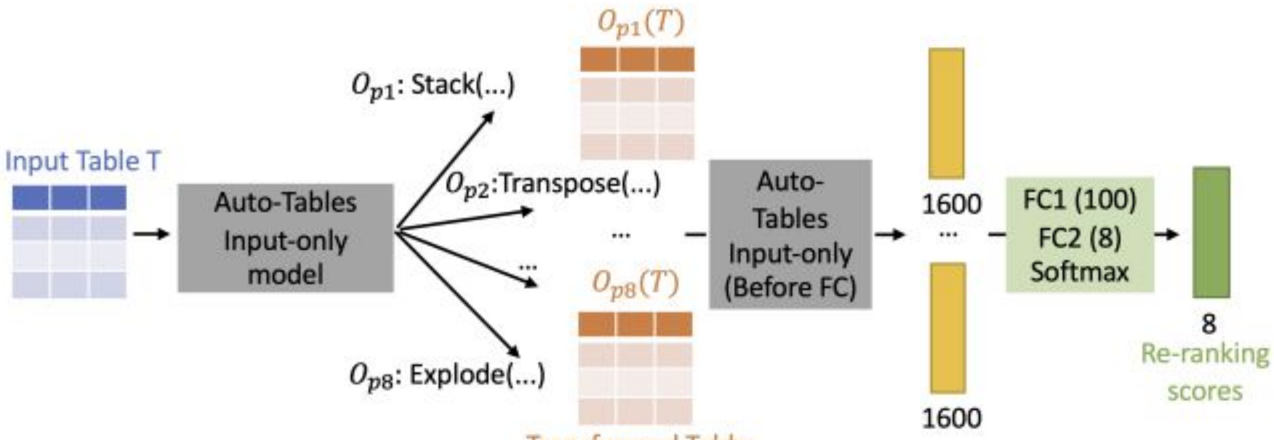
- We perform the beam search on the most likely top-k steps, to get the most likely operator and parameters sequence.

Input/Output Reranking

- Challenge: sometimes input characteristics alone are not sufficient to predict the best transformation
- Solution: Use both input table *T* and output table *M(T)* to re-rank transformations to predict the best transformation

Input/Output Reranking Model

- **Step 1**: Input-only synthesis model generates a set of top-k likely transformations.
- **Step 2**: For each transformation, apply it to the input table *T* to generate the output tables.
- **Step 3**: Convert each output table into a feature vector (using embedding and feature extraction).
- **Step 4**: Concatenate feature vectors of all top-k transformations and use fully connected layers to generate re-ranking scores.



Experiments

- Performed extensive evaluation of the algorithms using real test data
- **Experimental Setup:**
    - **Data Sources:**
        - Forums, Jupyter Notebooks, Excel/Web Tables
    - **Benchmark:**
        - Total of 244 test cases (26 require multi-step transformation)
        - Each case has an input table, the ground-truth transformation, and the expected output table that is relational

Evaluation

Quality: Hit@K

$$Hit@k(T) = \sum_{i=1}^{k} \mathbf{1}(\hat{M}_i(T) = M_g(T))$$

Efficiency: Latency of synthesis using wall-clock time

Results

Table 3: Quality comparison using Hit@k, on 244 test cases

Method	No-example methods				By-example methods			
	Auto-Tables	TaBERT	TURL	GPT-3.5-fs	FF	FR	SQ	SC
Hit @ 1	**0.570**	0.193	0.029	0.196	0.283	0.336	0	0
Hit @ 2	**0.697**	0.455	0.071	-	-	-	0	0
Hit @ 3	**0.75**	0.545	0.109	-	-	-	0	0
Upper-bound	-	-	-	-	0.471	0.545	0.369	0.369

Different table representations

SQL-by-example

Results

Table 4: Synthesis latency per test case

Method	Auto-Tables	Foofah (excl. 110 timeout cases)	FlashRelate (excl. 91 timeout cases)
50 %tile	**0.127s**	0.287s + human effort	3.4s + human effort
90 %tile	**0.511s**	22.891s + human effort	57.16s + human effort
95 %tile	**0.685s**	39.188s + human effort	348.6s + human effort
Average	**0.224s**	5.996s + human effort	59.194s + human effort

Results

Table 5: Ablation Studies of AUTO-TABLES

Method	Full	No Re-rank	No Re-rank &				
			No Aug	No Bert	No Syn	1x1 Only	5x5
Hit@1	**0.570**	0.508	0.463	0.467	0.504	0.471	0.480
Hit@2	**0.697**	0.652	0.582	0.627	0.648	0.607	0.594
Hit@3	**0.75**	0.730	0.656	0.693	0.676	0.652	0.660

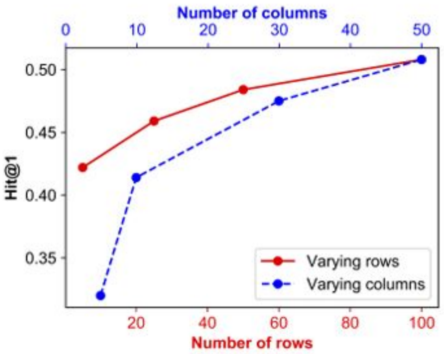


Figure 11: Vary input size

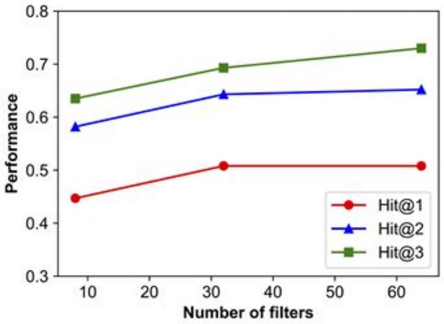


Figure 12: Vary number of filters

Results

Table 6: Sensitivity to different semantic embeddings.

Embedding methods	sentenceBERT	fastText	GloVe	No Semantic
Hit@1	0.508	0.529	0.525	0.467
Hit@2	0.652	0.656	0.676	0.627
Hit@3	0.730	0.734	0.734	0.734
Avg. latency per-case w/ this embedding	0.299s	0.052s	0.050s	0.026s

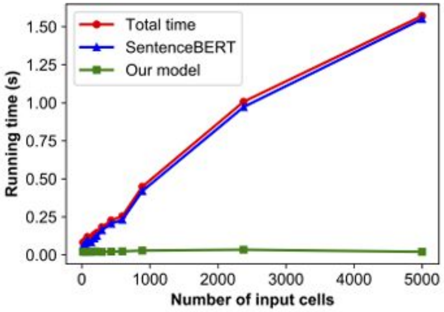


Figure 10: AUTO-TABLES latency analysis

Related work

- By-example transformation using program synthesis
  - “Row-to-row” transformations (e.g. TDE and FlashFill)
  - “Table-to-table” transformations (e.g., Foofah, PATSQL, QBO, and Scythe)
  - Orthogonal to Auto-tables
- Computer vision models for object detection
  - Algo in Auto-Tables inspired by CNN-architectures for object detection
  - But specifically designed for table transformation task.
- Representing tables using deep models
  - E.g., TaBERT, Tapas, Turl, etc.
  - Focus on natural-language (NL) aspects of tables, and tailor to NL-related tasks
  - Not suited for table-transformation task
- Database schema design
  - Decompose one large table into multiple smaller tables (3NF, BCNF, etc.)
  - Reconstruction in Auto-Tables is always single-table to single-table

Conclusion

Auto-Tables:

- Synthesize transformations to relationalize tables
- Use compute-vision-inspired algorithms
- Obviate the need for users to provide input/output examples
- Efficient and fast

Future Work:

- Extend the functionality to a broader set of operators
- Explore the applicability of this technique on other classes of transformations.

Study Questions

1. How does Auto-Tables' use of self-supervision and computer vision techniques contribute to its ability to transform tables without requiring user examples?
2. What are the key challenges in transforming non-relational tables to relational formats, and how does Auto-Tables address these challenges compared to traditional methods?