

TURL : Table Understanding through Representation Learning

<https://arxiv.org/pdf/2006.14806>

0. Introduction



- 웹 테이블에는 많은 구조적 지식이 존재하지만, 일반 언어 모델은 테이블 구조를 잘 반영하지 못함.
- 기존 접근은 셀의 텍스트만 사용하는 방식이 많아, 행·열·헤더 등 구조적 관계를 활용하지 못하는 한계가 있었음.
- 논문은 대규모 Wikipedia 테이블을 기반으로 사전학습한 TURL(Table Representation Learning) 모델을 제안함.
- 핵심 기여
 - 테이블 구조를 고려한 embedding 프레임워크
 - self-supervised pretraining objectives 적용
 - 다섯 가지 테이블 이해 태스크에서 통합적으로 활용 가능함

1. Overview

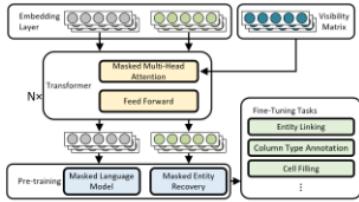


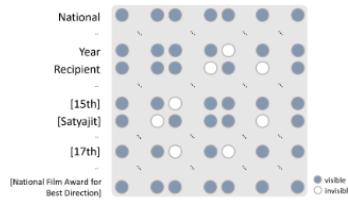
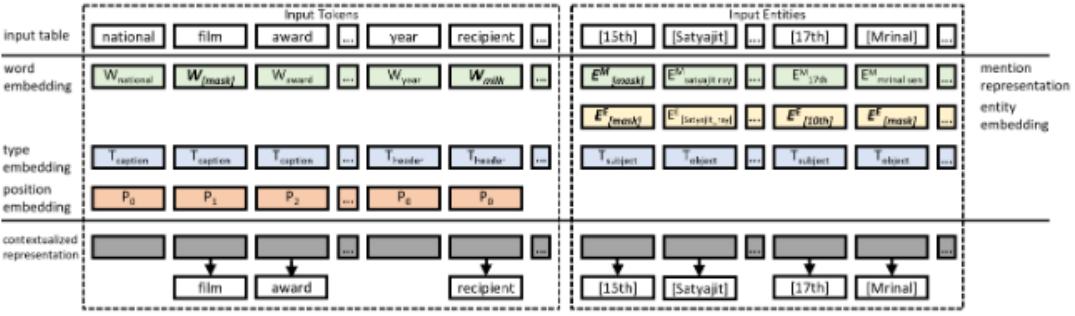
Figure 2: Overview of our TURL framework.

- TURL은 테이블의 셀, 행, 열, 헤더 정보를 통합적으로 인코딩하는 모델 구조를 가짐.
- 테이블을 그래프 구조로 보고, 각 셀과 엔티티를 transformer encoder에서 처리함.
- Pretraining은 Wikipedia 테이블 코퍼스 기반으로, entity recovery·column type prediction 등의 self-supervised 목표를 포함함.
- 적용 범위는 table understanding, knowledge base 확장, semantic parsing 등 임.

2. Challenges

- 웹 테이블은 문맥이 부족하고 텍스트가 짧아 기존 language model로는 의미 파악이 어려움.
- column header가 불완전·모호한 경우가 많아 column type 추론이 어렵고, 동일 엔티티도 테이블마다 표현이 달라 linking 난이도가 높음.
- 테이블 구조가 매우 다양해 일관된 구조 학습이 어려움.
- 대규모 구조적 테이블 데이터셋이 부족해 robust한 모델 학습에 제약이 존재했음.

3. Method



Task		Finetune Strategy
Table Interpretation	Entity Linking	[national film year recipient 1967[15th] Satyajit]
		[15th] [Satyajit]
	Column Type Annotation	[national film year recipients 15th [15th] [Satyajit]]
		event person, actor
	Relation Extraction	[national film year recipient 15th [15th] [Satyajit]]
Table Augmentation	Row Population	[national film year recipient 15th [15th] [Satyajit]]
		[16th], [17th], [18th]
	Cell Filling	[national film year recipient 15th [15th] [MASK]]
		[Satyajit]
	Schema Augmentation	[national film award year recipient 15th [MASK]]

- 입력 테이블을 flatten하지 않고 row, column, cell, header 단위로 구조 정보를 유지 함.
- Transformer encoder 기반으로, 테이블 전용 positional encoding을 추가해 구조적 관계를 반영함.
- 셀 임베딩은 token embedding + entity embedding으로 구성됨.

- Pretraining objectives
 - entity recovery
 - masked entity modeling
 - column type prediction
 - row/column relation reasoning
- 최적화는 Adam 기반인데, 세부 하이퍼파라미터는 논문 PDF 외에는 “확실하지 않음”

4. Experiments

	split	min	mean	median	max
# row	train	1	13	8	4670
	dev	5	20	12	667
	test	5	21	12	3143
# ent. columns	train	1	2	2	20
	dev	3	4	3	15
	test	3	4	3	15
# ent.	train	3	19	9	3911
	dev	8	57	34	2132
	test	8	60	34	9215

- 데이터
 - Wikipedia 약 7M 테이블 기반 사전학습 데이터 구축 (논문 기준 확실함)
 - Wikidata entity ID를 기반으로 entity-linked table 구성
- 비교 baseline
 - TabMlp, TabVec, BERT 기반 모델
- 평가 지표
 - accuracy, F1 등 태스크별 알맞은 지표 사용
- 평가 태스크
 - Cell entity linking
 - Column type prediction
 - Column relations
 - Row population
 - Entity prediction

5. Results

Table 6: Model evaluation on column type annotation task.

Method	F1	P	R
Sherlock (only entity mention) [21]	78.47	88.40	70.55
TURL + fine-tuning (only entity mention)	88.86	90.54	87.23
TURL + fine-tuning	94.75	94.95	94.56
w/o table metadata	93.77	94.80	92.76
w/o learned embedding	92.69	92.75	92.63
only table metadata	90.24	89.91	90.58
only learned embedding	93.33	94.72	91.97

Table 7: Accuracy on T2D-Te and Efthymiou, where scores for HNN + P2Vec are copied from [11] (trained with 70% of T2D and Efthymiou respectively and tested on the rest). We directly apply our models by type mapping without retraining.

Method	T2D-Te	Efthymiou
HNN + P2Vec (entity mention + KB) [11]	0.966	0.865
TURL + fine-tuning (only entity mention) + table metadata	0.888 0.860	0.745 0.904

Table 8: Accuracy on T2D-Te and Efthymiou. Here all models use T2D-Tr (70% of T2D) as training set, following the setting in [11].

Method	T2D-Te	Efthymiou
HNN + P2Vec (entity mention + KB) [11]	0.966	0.650
TURL + fine-tuning (only entity mention) + table metadata	0.940 0.962	0.516 0.746

Table 9: Further analysis on column type annotation: Model performance for 5 selected types. Results are F1 on validation set.

Method	person	pro_athlete	actor	location	citytown
Sherlock	96.85	74.39	29.07	91.22	55.72
TURL + fine-tuning	99.71	91.14	74.85	99.32	79.72
only entity mention	98.44	87.11	58.86	96.59	60.13
w/o table metadata	99.63	90.38	74.46	99.01	77.37
w/o learned embedding	99.38	90.56	71.39	98.91	75.55
only table metadata	98.26	88.80	70.86	98.11	72.54
only learned embedding	98.72	91.06	73.62	97.78	75.16

Table 10: Model evaluation on relation extraction task.

Method	F1	P	R
BERT-based	90.94	91.18	90.69
TURL + fine-tuning (only table metadata)	92.13	91.17	93.12
TURL + fine-tuning	94.91	94.57	95.25
w/o table metadata	93.85	93.78	93.91
w/o learned embedding	93.35	92.90	93.80

Table 11: Relation extraction results of an entity linking based system, under different agreement ratio thresholds.

Min Ag. Ratio	F1	P	R
0	68.73	60.33	79.85
0.4	82.10	94.65	72.50
0.5	77.68	98.33	64.20
0.7	63.10	99.37	46.23

Table 12: Model evaluation on row population task. Recall is the same for all methods because they share the same candidate generation module.

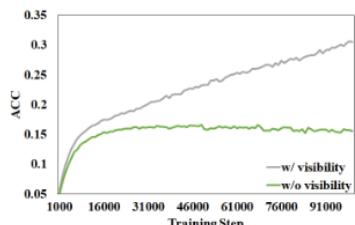
# seed	0		1	
Method	MAP	Recall	MAP	Recall
EntiTables [45]	17.90	63.30	42.31	78.13
Table2Vec [13]	-	63.30	20.86	78.13
TURL + fine-tuning	40.92	63.30	48.31	78.13

Table 13: Model evaluation on cell filling task.

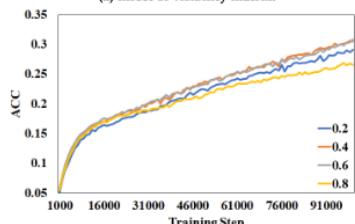
Method	P @ 1	P @ 3	P @ 5	P @ 10
Exact	51.36	70.10	76.80	84.93
H2H	51.90	70.95	77.33	85.44
H2V	52.23	70.82	77.35	85.58
TURL	54.80	76.58	83.66	90.98

Table 14: Model evaluation on schema augmentation task.

Method	#seed column labels	
	0	1
kNN	80.16	82.01
TURL + fine-tuning	81.94	77.55



(a) Effect of visibility matrix.



(b) Effect of different MER mask ratios.

Figure 7: Ablation study results.

- 모든 테이블 이해 태스크에서 기존 baseline보다 더 높은 성능을 기록함.
- 특히 entity linking과 column type inference에서 큰 개선을 보임.
- 구조적 positional encoding이 성능에 결정적이며, ablation 결과에서 제거 시 정확도가 크게 감소함.
- 세부 파라미터 크기 및 training cost는 “확실하지 않음”.

6. Insight

- 테이블은 자연어와 다른 구조적 특징을 가지므로, 테이블 전용 구조 학습이 필요하다는 점을 입증함.
- 대규모 테이블 기반 사전학습이 downstream table reasoning 성능 향상에 효과적임.
- Knowledge base 확장, table QA 등 다양한 응용 가능성을 보여줌.
- 그러나 숫자 중심 테이블이나 구조가 지나치게 특이한 테이블에서는 성능 제한이 있을 수 있음(추측입니다).
- 향후 가능성
 - 복합형 테이블(multimodal table) 처리
 - spreadsheet·DB 등 비정형 테이블로 확장
 - LLM 기반 reasoning과 결합한 hybrid table model 연구