ColNet: Embedding the Semantics of Web Tables for Column Type Prediction

Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, Charles Sutton

Introduction

Column Type Prediction

- **Problem**
  - Matching an entity column, whose cells are text phrases (i.e., entity mentions), with classes of a knowledge base (KB)
- **Example**
  - A column composed of “Mute swan”, “Yellow-billed duck” and “Wandering albatross”
  - DBPedia classes dbo:Species, dbo:Bird
- **Importance**
  - The base of column relation annotation, foreign key discovery, etc.
  - Data analytics, interpretable machine learning, etc.

Difficulty in Column Type Prediction

- Multiple and hierarchical classes
- Identifying a fine-grained class (dbo:BasketballPlayer VS dbo:Athlete VS dbo:Person)
- Column cells may have few or even empty KB entity correspondences, which is referred to as *knowledge gap*
- Disambiguation, e.g., “Virgin” as “Mary” or as “Virgin Media”

Background

- **Collective approaches**
  - Probabilistic graph model [Limaye et al. VLDB'10][Mulwad et al. ISWC'13][Bhagavatula et al. ISWC'15]
  - Iterative: TableMiner+ [Zhang SWJ 2017], T2K Match [Ritze et al. WIMS'15]

Background

- **Collective approaches**
  - Probabilistic graph model [Limaye et al. VLDB'10][Mulwad et al. ISWC'13][Bhagavatula et al. ISWC'15]
  - Iterative: TableMiner+ [Zhang SWJ 2017], T2K Match [Ritze et al. WIMS'15]
- **Cell to entity matching + Voting**
  - Lookup with lexical index (e.g., `http://lookup.dbpedia.org`)
  - Machine learning for contextual semantics [Efthymiou et al. ISWC'17][Luo et al. AAAI'18]

Background

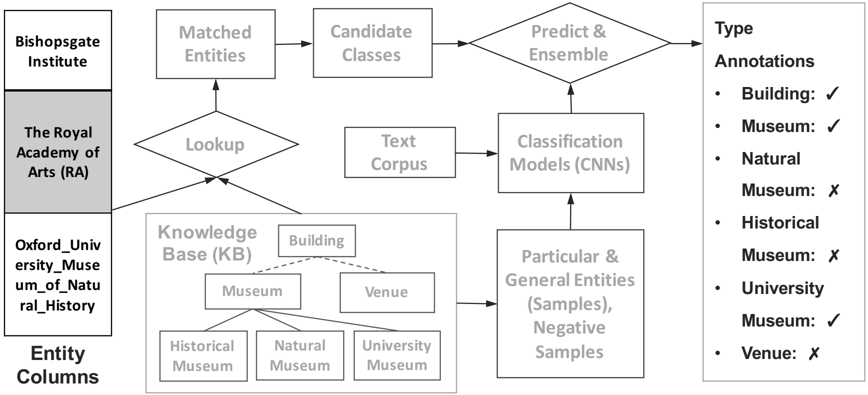
- **Collective approaches**
  - Probabilistic graph model [Limaye et al. VLDB'10][Mulwad et al. ISWC'13][Bhagavatula et al. ISWC'15]
  - Iterative: TableMiner+ [Zhang SWJ 2017], T2K Match [Ritze et al. WIMS'15]
- **Cell to entity matching + Voting**
  - Lookup with lexical index (e.g., `http://lookup.dbpedia.org`)
  - Machine learning for contextual semantics [Efthymiou et al. ISWC'17][Luo et al. AAAI'18]
- **Knowledge gap**
  - Reference columns [Pham et al. ISWC'16], search engine [Quercini et al. EDBT'13]

Methods

ColNet in a Nutshell

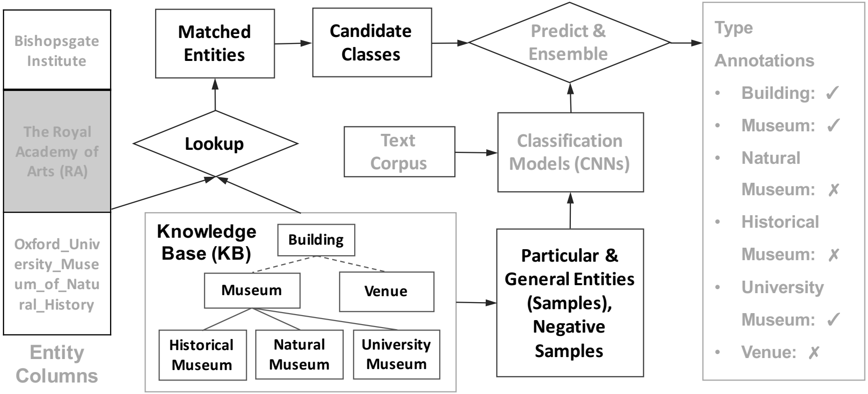
- Does **not** assume the existence of **metadata** (e.g., column headers)
- Uses a **Convolutional Neural Network** (CNN) for contextual semantics (features)
- Relies on KB lookup and SPARQL queries for **automatic sampling**
- Considers the **knowledge gap**
  - *i.e.*, a column has a small number of cells, or the cells have missing or not accurate KB entity correspondences

Overview



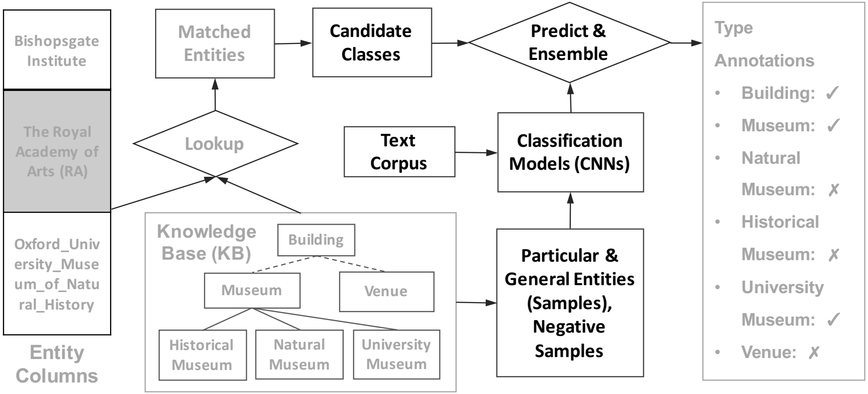
1. Example
2. Lookup & Query
3. Prediction

Overview



1. Example
2. Lookup & Query
3. Prediction

Overview



1. Example
2. Lookup & Query
3. Prediction

KB entity matching with Lookup and Query

- **Lookup & Query**
  - Lexical index based on entity label and entity anchor text
    - Cells $\rightarrow$ entity correspondences
  - SPARQL Query
    - Entity correspondences $\rightarrow$ candidate classes
- **One-vs-rest**
  - For one candidate class, we train one binary classifier (CNN)

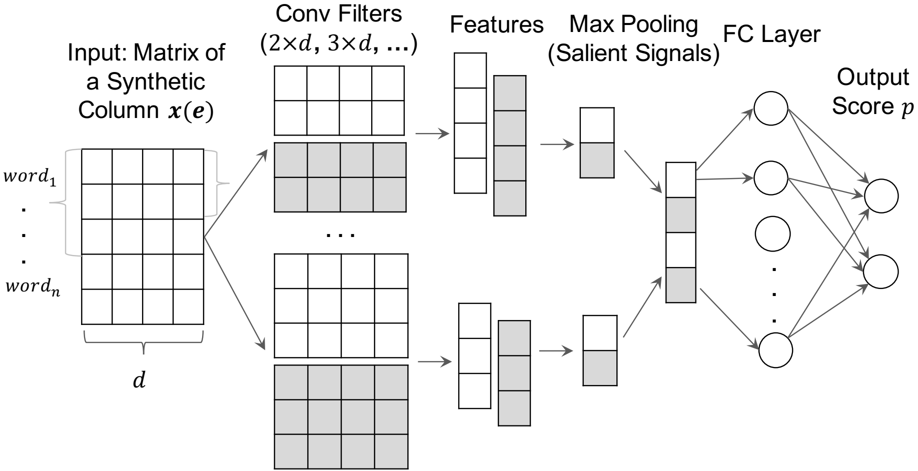
Synthetic Columns (I)

- They are our (positive and negative) **samples**
- In **training** they are composed by a subsets of KB entities
- In **prediction** they are segments of the column
- Synthetic columns are **embedded into a matrix** by stacking the word vectors of the involved entities

Synthetic Columns (II)

- **Benefit**: learn inter-cell correlations (locality features) by CNN
- **Example**:
  - (“Apple", “MS", “Google") as column.
  - Expected prediction of “IT company" as column type
  - Prediction cell by cell will probably have a score from 0.33 to 0.66
  - Prediction score considering the correlation between cells $\approx$ 1.0

CNN architecture in ColNet



Sampling for a candidate class

- A sample is composed by a **set of entities** (with the size of the synthetic column) and the **class label**
- **Positive sampling**
  - **Particular samples**: KB entities that are matched with column cells
  - **General samples**: common KB entities that are instance-of the class

Sampling for a candidate class

- A sample is composed by a **set of entities** (with the size of the synthetic column) and the **class label**
- **Positive sampling**
  - **Particular samples**: KB entities that are matched with column cells
  - **General samples**: common KB entities that are instance-of the class
- **Negative samples** are composed by matched entities of other candidate classes that
  - are disjoint with the class (e.g., entities of “Fruit" wrt “IT Company")
  - appear together with the class in at least one column
- Balanced positive and negative samples

Training with Transfer learning

- Two Steps:
  - CNN **pre-training** with **general samples**
  - CNN **fine-tuning** with **particular samples**

- Benefits
  - Deal with the shortage of particular samples (the knowledge gap)
  - Bridge the data distribution gap between KB entities and table cells

Ensemble

- s^c: ensemble score for class c

$$s^c = \begin{cases} v^c, & \text{if } v^c \geq \sigma_1 \text{ or } v^c < \sigma_2 \\ p^c, & \text{otherwise} \end{cases}$$

- v^c: voting score computed as the rate of cells linked to the class c.
- p^c: average of the scores predicted by the CNNs for each synthetic column.

Ensemble

- s^c: ensemble score for class c

$$s^c = \begin{cases} v^c, & \text{if } v^c \geq \sigma_1 \text{ or } v^c < \sigma_2 \\ p^c, & \text{otherwise} \end{cases}$$

- v^c: voting score computed as the rate of cells linked to the class c.
- p^c: average of the scores predicted by the CNNs for each synthetic column.
- Benefits:
  - Classes *voted* by the majority of cells typically have high precision.
  - The CNN-based prediction model focuses on the cases where there is ambiguity in the matched entities or a significant knowledge gap.

Evaluation

Evaluation setting: data

- **DBPedia** as the KB
- Word embedding: **Word2vec** model trained with the latest dump of Wikipedia pages
- **T2Dv2** (tables from the Web) and **Limaye** (tables from Wikipedia pages) datasets
- Limaye dataset more challenging in terms of **knowledge gap**

Name	Columns	Avg. Cells	Different “Best” (“Okay”) Classes
T2Dv2	411	124	56 (35)
Limaye	428	23	21 (24)

Evaluation setting: ground truth and baselines

- **Evaluation models**
  - “Best" and “okay" ground truth classes
  - “Strict" and “tolerant" evaluation models
- **Baselines**
  - DBPedia Lookup + Vote
  - T2K Match
  - Efthymiou et al. 2017 + Vote

Our methods: ColNet and $\text{ColNet}_{\text{Ensemble}}$

Overall Results on Limaye

Precision, Recall, F1 score

Models	Methods	PK Columns
Tolerant	$\text{ColNet}_{\text{Ensemble}}$	**0.796**, 0.799, **0.798**
	ColNet	0.763, **0.820**, 0.791
	Lookup-Vote	0.732, 0.660, 0.694
	T2K Match	0.560, 0.408, 0.472
	Efthymiou17-Vote	0.759, 0.414, 0.536
Strict	$\text{ColNet}_{\text{Ensemble}}$	0.602, **0.639**, **0.620**
	ColNet	0.576, 0.619, 0.597
	Lookup-Vote	0.571, 0.447, 0.501
	T2K Match	0.453, 0.330, 0.382
	Efthymiou17-Vote	**0.626**, 0.357, 0.454

- **Prediction impact**
  - $\text{ColNet}_{\text{Ensemble}}$ and ColNet > Lookup-Vote
  - Improve recall dramatically
- **Ensemble impact**
  - $\text{ColNet}_{\text{Ensemble}}$ > ColNet
  - Improves precision dramatically
- **Comparison with the state-of-the-art**
  - $\text{ColNet}_{\text{Ensemble}}$ and ColNet > T2K Match
  - $\text{ColNet}_{\text{Ensemble}}$ and ColNet has competitive precision as Efthymiou17-Vote, but much higher recall and F1 score

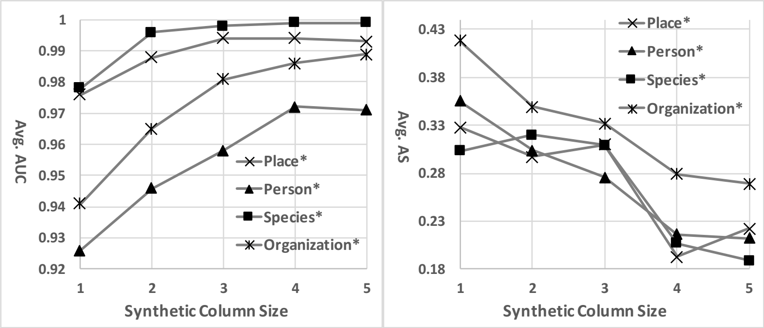
Overall Results on T2Dv2

Precision, Recall, F1 score

Models	Methods	All Columns	PK Columns
Tolerant	$\text{ColNet}_{\text{Ensemble}}$	**0.917**, **0.909**, **0.913**	**0.967**, **0.985**, **0.976**
	ColNet	0.845, 0.896, 0.870	0.927, 0.960, 0.943
	Lookup-Vote	0.909, 0.865, 0.886	0.965, 0.960, 0.962
	T2K Match	0.664, 0.773, 0.715	0.738, 0.895, 0.809
Strict	$\text{ColNet}_{\text{Ensemble}}$	0.853, **0.846**, **0.849**	0.941, **0.958**, **0.949**
	ColNet	0.765, 0.811, 0.787	0.868, 0.898, 0.882
	Lookup-Vote	**0.862**, 0.821, 0.841	**0.946**, 0.941, 0.943
	T2K Match	0.624, 0.727, 0.671	0.729, 0.884, 0.799

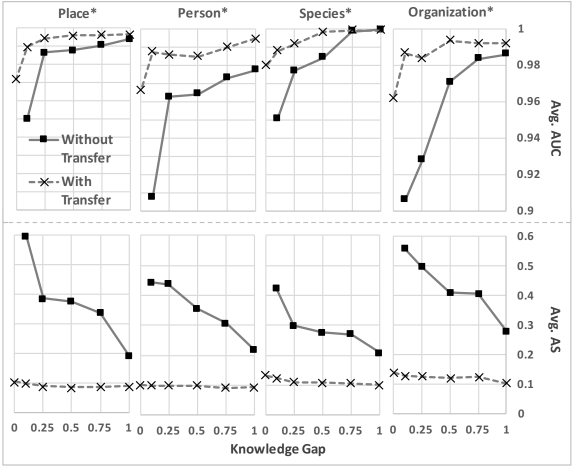
- Prediction impact
- Ensemble impact
- Ensemble impact
- **Knowledge gap impact**
  - Limaye is harder than T2Dv2 although it has less “best” and “okay” classes
  - Limaye has shorter columns in average, which causes larger knowledge gap
  - Improvements of $\text{ColNet}_{\text{Ensemble}}$ and ColNet on Limaye are more significant, due to that ColNet deals with knowledge gap

Impact of synthetic column size on CNNs



The testing performance of CNNs on Truly Matched classes [left] and Falsely Matched classes [right] for types of columns: Place, Person, Species & Organization. AUC: area under ROC curve, AS: average score

Impact of transfer learning and the knowledge gap on CNNs



- The testing performance of CNNs of TM classes [above] and FM classes [below]
  - under different knowledge gaps
  - with and without transfer learning
  - Four types of columns: Place, Person, Species and Organization
- The knowledge gap is simulated by randomly selecting a ratio of particular entities for training. The lower ratio, the larger gap.

Conclusions and Future Work

Conclusions

We present a column type prediction framework named **ColNet** that

- utilizes CNNs, semantic embedding and KBs.
- does not assume the existence of table metadata
- learns both cell level and column level semantics
- automatically trains prediction models utilizing KB lookup and reasoning
- uses transfer learning to address the knowledge gap
- outperforms state-of-the-art approaches when columns entities are scarce

Future work

– Learning stronger table locality features (contextual semantics)

– Use ColNet as the basis for other Web table to KB matching tasks

– Application of ColNet in the data science pipeline as an AI assistant
  – project: Artificial Intelligence for Data Analytics
    `https://www.turing.ac.uk/research/research-projects/artificial-intelligence-data-analytics`

Questions?

Main contacts:
Jiaoyan Chen (`jiaoyan.chen@cs.ox.ac.uk`)
Ernesto Jimenez Ruiz (`ejimenez-ruiz@turing.ac.uk`)

Sources, datasets, paper and slide:

`https://github.com/alan-turing-institute/SemAIDA/`