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

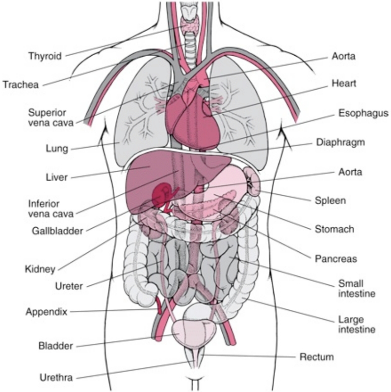
Jiaoyan Chen, *Ernesto Jiménez-Ruiz*, Ian Horrocks, Charles Sutton

The Thirty-Third AAAI Conference on Artificial Intelligence (**AAAI-19**)

Preliminaries

What is an ontology?

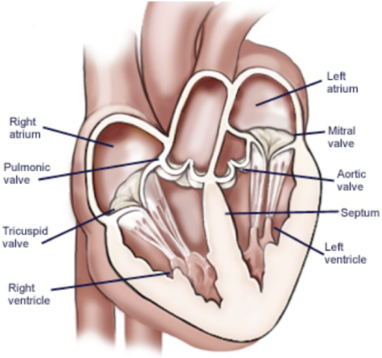
- Introduces **vocabulary** relevant to a domain
  - Anatomy



(*) Borrowed from Ian Horrocks' slides: **Ontologies and the Semantic Web: The Story So Far**. April 2010

What is an ontology?

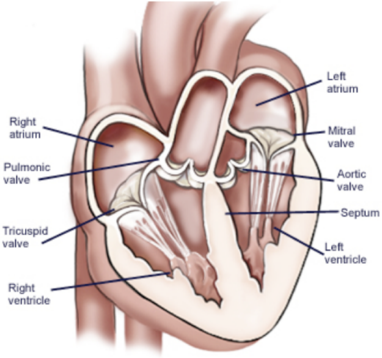
- Specifies meaning (**semantics**) of terms
  - Heart is a muscular organ that is part of the circulatory system



(*) Borrowed from Ian Horrocks' slides: **Ontologies and the Semantic Web: The Story So Far**. April 2010

What is an ontology?

- Specifies meaning (**semantics**) of terms
  - Heart is a muscular organ that is part of the circulatory system
- **Formalised** using suitable logic
  - Heart SUBCLASSOF MuscularOrgan AND (isPartOf SOME CirculatorySystem)



(*) Borrowed from Ian Horrocks' slides: **Ontologies and the Semantic Web: The Story So Far**. April 2010

What ontologies are good for?

- Independence of logical/physical schema: **domain model**
- Vocabulary closer to domain experts: **more user-friendly**
- Incomplete and semi-structured data: **flexibility**
- Integration of heterogeneous sources: **unified view**

What is a Knowledge Graph (KG)?

– “Large network of entities, their semantic types, properties and relationships between entities.” (JWS Special Issue on Knowledge Graphs)

What is a Knowledge Graph (KG)?

- “Large network of entities, their semantic types, properties and relationships between entities.” (JWS Special Issue on Knowledge Graphs)
- **(Light) Knowledge Base**: with a (light) terminology (ontology) and assertions (data)
- Nicer name than **RDF graph** (Resource Description Framework)
- Examples: Google Knowledge Graph, DBpedia (KG version of Wikipedia)

Overview: Role of Semantics in AIDA

Semantics for Data Analytics

– The **lack of semantics and context in datasets** hinders the application of data analysis tools to, for example, identify errors like wrong values.

Semantics for Data Analytics

- The **lack of semantics and context in datasets** hinders the application of data analysis tools to, for example, identify errors like wrong values.
- **Ontologies** model the domain of application (e.g., expected cardinalities, relationships, accepted range of values for a *temperature sensor*).
- **Rules** to identify potential missing data (e.g., a person must have a name).

Semantics for Data Analytics

- The **lack of semantics and context in datasets** hinders the application of data analysis tools to, for example, identify errors like wrong values.
- **Ontologies** model the domain of application (e.g., expected cardinalities, relationships, accepted range of values for a *temperature sensor*).
- **Rules** to identify potential missing data (e.g., a person must have a name).
- Ontologies and rules **to validate new learned knowledge**.
- Use of a shared **semantic store** among data analysis tools (**Semantics-aware AI assistants**)

Adding semantics to Tabular Data

- Assigning a semantic type (e.g., a KG class) to an (entity) column
- Matching a cell to a KG entity
- Assigning a KG property to the relationship between two columns

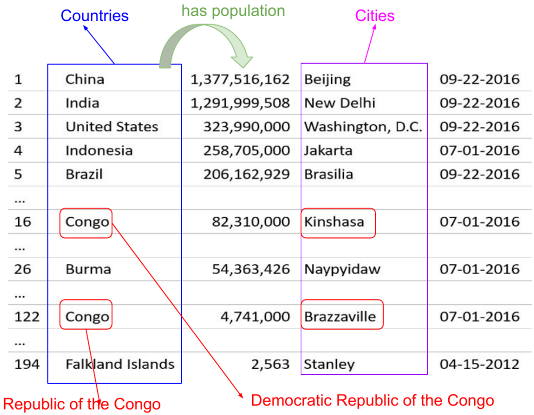
(*) We assume the existence of a (possibly incomplete) **Knowledge Graph (KG)** relevant to the domain.

Adding semantics to Tabular Data

- **Assigning a semantic type (e.g., a KG class) to an (entity) column**
- Matching a cell to a KG entity
- Assigning a KG property to the relationship between two columns

(*) We assume the existence of a (possibly incomplete) **Knowledge Graph (KG)** relevant to the domain.

Adding semantics to Tabular Data


Countries
has population
Cities
Republic of the Congo
Democratic Republic of the Congo

1	China	1,377,516,162	Beijing	09-22-2016
2	India	1,291,999,508	New Delhi	09-22-2016
3	United States	323,990,000	Washington, D.C.	09-22-2016
4	Indonesia	258,705,000	Jakarta	07-01-2016
5	Brazil	206,162,929	Brasilia	09-22-2016
...				
16	Congo	82,310,000	Kinshasa	07-01-2016
...				
26	Burma	54,363,426	Naypyidaw	07-01-2016
...				
122	Congo	4,741,000	Brazzaville	07-01-2016
...				
194	Falkland Islands	2,563	Stanley	04-15-2012

(*) Adapted from Efthymiou et al. Matching Web Tables with Knowledge Base Entities: From Entity Lookups to Entity Embeddings. ISWC 2017

Contribution of Semantics in Data Wrangling Challenges

- *Data parsing*, e.g. converting csv's or tables.
- (+++)*Data dictionary*: basic types and semantic types.
- (++)*Data integration* from multiple sources (foreign key discovery).
- (++)*Entity resolution*: duplication and record linkage.
- *Format variability*: e.g. for dates and names.
- (+)*Structural variability* in the data.
- (++)Identifying and repairing *missing data*.
- (+)*Anomaly detection* and repair.
- (+++)**Metadata/contextual information**. (Semantic) data governance.

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

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

The Thirty-Third AAAI Conference on Artificial Intelligence (**AAAI-19**)

Introduction

Challenges 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 KG entity correspondences, which is referred to as **knowledge gap**
- **Disambiguation**, e.g., “Virgin” as “Mary” or as “Virgin Media”

Methods

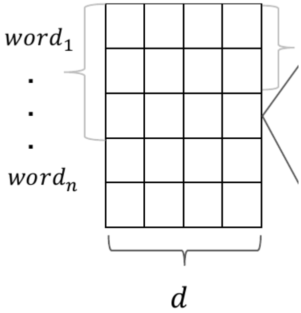
ColNet in a Nutshell

- utilizes **Convolutional Neural Networks** (**CNNs**), **semantic embeddings** and **Knowledge Graphs**.
- does **not** assume the existence of table **metadata**
- learns both **cell level** and **column level semantics**
- **automatically trains** prediction models relying on a Knowledge Graph
- uses **transfer learning** to address the knowledge gap
- **outperforms state-of-the-art** approaches when column entities are scarce

Samples and embeddings

- The **CNNs** expect a matrix as input.
- **Semantic embeddings**: low-dimensional (vector space) representation of words.
- (Positive and negative) **samples** as a stack of word vectors
- In training, samples are **automatically labelled** with a KG class.

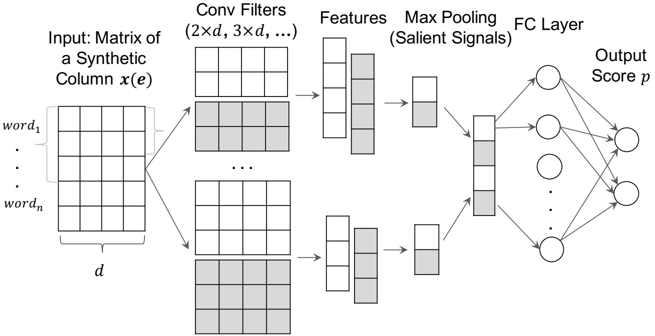
Input: Matrix of a Synthetic Column $\boldsymbol{x}(\boldsymbol{e})$



Training in ColNet

ColNet trains a CNN for each (candidate) KG class.

1. **pre-trains** the CNN with (general) samples from the KG,
2. **fine tunes** the CNN with (particular) samples from the table column.



Input: Matrix of a Synthetic Column $\boldsymbol{x}(\boldsymbol{e})$
$word_1$
$word_n$
d
Conv Filters
$(2\times d,\ 3\times d,\ \ldots)$
Features
Max Pooling
(Salient Signals)
FC Layer
Output Score p

Pre-training: (general) samples from the Knowledge Graph

- We use **members/entities of the class in the KG**
  - For example (DBPedia): "dbr:Apple_Inc", "dbr:Microsoft", "dbr:Google", "dbr:Amazon.com" and "dbr:Alibaba Group" are members of the class "dbo:Company".

Pre-training: (general) samples from the Knowledge Graph

- We use **members/entities of the class in the KG**
  - For example (DBPedia): "dbr:Apple_Inc", "dbr:Microsoft", "dbr:Google", "dbr:Amazon.com" and "dbr:Alibaba Group" are members of the class "dbo:Company".
- **A sample** (or synthetic column) is built by grouping a specific number of entities.
  - For example "dbr:Amazon" and "dbr:Alibaba Group" would form a sample of size 2 for the class "dbo:Company".
  - Matrix representation (stack of the word vectors): v(“Amazon”) $\oplus$ v(“Alibaba”) $\oplus$ v(“Group”)
- The **size of the sample** is one of our hyper-parameters.

Fine-Tuning: (particular) samples from table column

- **KG Look-up:** Lexical index based on entity labels
  - Cells → KG entity
  - e.g., Apple → "dbr:Apple", "dbr:Apple_Inc"

Column X
Apple
MS
Google

Fine-Tuning: (particular) samples from table column

- **KG Look-up:** Lexical index based on entity labels
  - Cells → KG entity
  - e.g., Apple → "dbr:Apple", "dbr:Apple_Inc"
- **KG Query:**
  - Entity → KG classes
  - e.g., "dbr:Apple" → "dbo:Fruit"
  - e.g., "dbr:Apple_Inc" → "dbo:Company"

Column X
Apple
MS
Google

Fine-Tuning: (particular) samples from table column

- **KG Look-up:** Lexical index based on entity labels
  - Cells → KG entity
  - e.g., Apple → "dbr:Apple", "dbr:Apple_Inc"
- **KG Query:**
  - Entity → KG classes
  - e.g., "dbr:Apple" → "dbo:Fruit"
  - e.g., "dbr:Apple_Inc" → "dbo:Company"

Column X
Apple
MS
Google

- **Sample generation:** segments of the column that are "dbo:Company".

Negative samples for a KG class (training)

- **Balanced** positive and negative **samples**
- Source of (**general**) negative samples:
  - * Exploits non members (especially from disjoint classes)

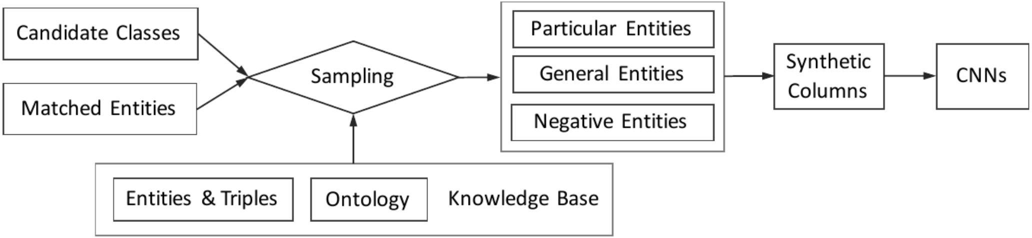
Negative samples for a KG class (training)

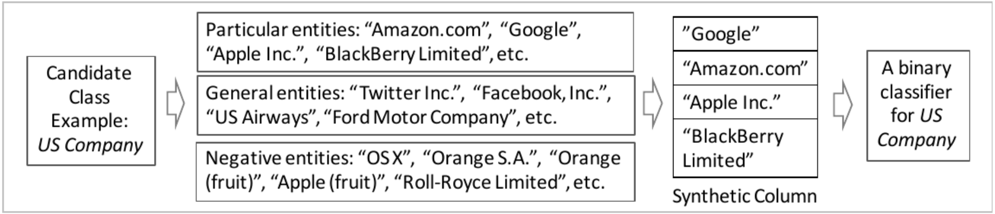
- **Balanced** positive and negative **samples**
- Source of (**general**) negative samples:
  - Exploits non members (especially from disjoint classes)
- Source of (**particular**) negative samples:
  - Entities that are disjoint with the KG class and appear together in the column
  - e.g., members of "dbo:Fruit" like "dbr:Apple"

Training with transfer learning in ColNet

- Recall two steps CNN training: **pre-training** and **fine-tuning**
- **Benefits**:
  - **Pre-training**: deals with the shortage of particular samples: knowledge gap or short columns
  - **Fine-tuning**: Bridge the data distribution gap between KG entities and table cells
- *[Impact analysis in the evaluation]*

Training and sampling end-to-end example




Candidate Class Example: *US Company*

Particular entities: “Amazon.com”, “Google”, “Apple Inc.”, “BlackBerry Limited”, etc.

General entities: “Twitter Inc.”, “Facebook, Inc.”, “US Airways”, “Ford Motor Company”, etc.

Negative entities: “OS X”, “Orange S.A.”, “Orange (fruit)”, “Apple (fruit)”, “Roll-Royce Limited”, etc.

”Google”
“Amazon.com”
“Apple Inc.”
“BlackBerry Limited”

Synthetic Column

A binary classifier for *US Company*

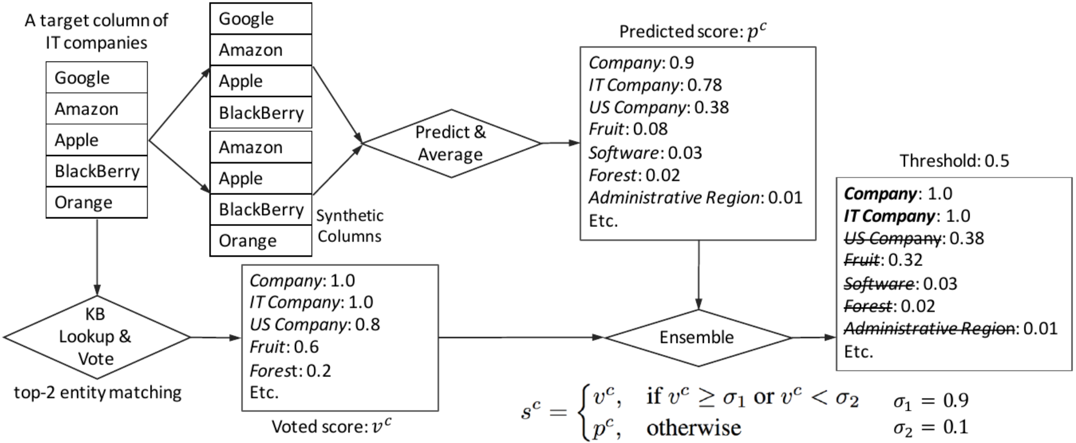
Prediction in ColNet

- **Prediction samples** are composed by segments of the column
- In our example: (“Apple", “MS", “Google") as column.
  - e.g. size 1: v(“Apple").
  - e.g. size 2: v(“Apple") $\oplus$ v(“Google").
  - e.g. size 3: v(“Apple") $\oplus$ v(“Google") $\oplus$ v(“MS").

Prediction in ColNet

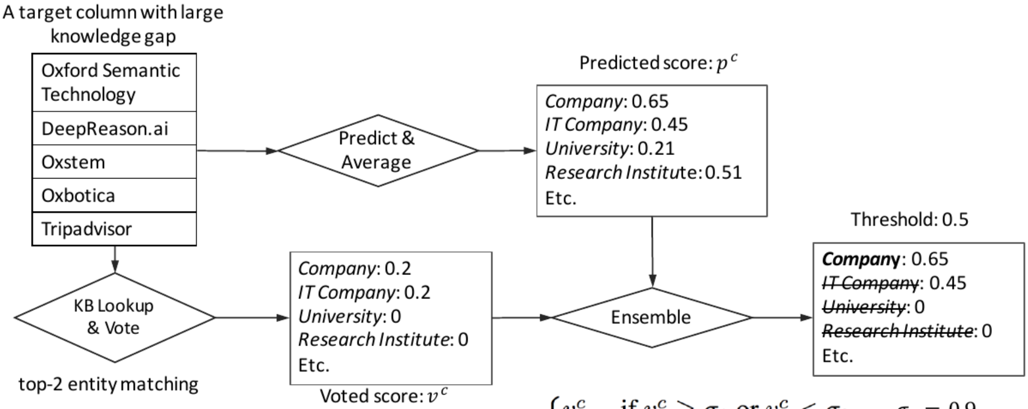
- **Prediction samples** are composed by segments of the column
- In our example: (“Apple", “MS", “Google") as column.
  - e.g. size 1: v(“Apple").
  - e.g. size 2: v(“Apple") $\oplus$ v(“Google").
  - e.g. size 3: v(“Apple") $\oplus$ v(“Google") $\oplus$ v(“MS").
- **Benefit of the sample size**: learn inter-cell correlations (locality features) by CNN
  - Expected prediction: “dbo:Company"
  - Prediction cell by cell: score from 0.33 to 0.66
  - Prediction score considering the correlation between cells $\approx$ 1.0
  - *[Impact analysis in the evaluation]*

Prediction in ColNet (Example 1)



$$s^c = \begin{cases} v^c, & \text{if } v^c \geq \sigma_1 \text{ or } v^c < \sigma_2 \\ p^c, & \text{otherwise} \end{cases} \qquad \begin{array}{l} \sigma_1 = 0.9 \\ \sigma_2 = 0.1 \end{array}$$

Prediction in ColNet (Example 2)



A target column with large knowledge gap

Oxford Semantic Technology
DeepReason.ai
Oxstem
Oxbotica
Tripadvisor

Predict & Average

Predicted score: p^c

Company: 0.65
IT Company: 0.45
University: 0.21
*Research Institu*te: 0.51
Etc.

KB Lookup & Vote

top-2 entity matching

Company: 0.2
IT Company: 0.2
University: 0
Research Institute: 0
Etc.

Voted score: v^c

Ensemble

Threshold: 0.5

Company: 0.65
~~*IT Company*~~: 0.45
~~*University*~~: 0
~~*Research Institute*~~: 0
Etc.

$$s^c = \begin{cases} v^c, & \text{if } v^c \geq \sigma_1 \text{ or } v^c < \sigma_2 \\ p^c, & \text{otherwise} \end{cases} \qquad \begin{aligned} \sigma_1 &= 0.9 \\ \sigma_2 &= 0.1 \end{aligned}$$

Evaluation

Evaluation setting: data

- **DBPedia** as the KG
- 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**
  - **Strict** (best hit only) and **tolerant** (compatible hits) evaluation models
- **Baselines**
  - DBPedia Lookup + Vote
  - T2K Match [Ritze et al. WIMS'15]
  - Efthymiou + Vote [Efthymiou et al. ISWC'17]
  - Other state of the art systems not available

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

ColNet 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 (prediction sample).

ColNet 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 (prediction sample).
- 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.

Overall Results on Limaye

Precision (P), Recall (R), F1 score (F)

Models	Methods	PK Columns		
		P	R	F
Tolerant	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	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**
  - ColNet$_{\text{Ensemble}}$ and ColNet > Lookup-Vote
  - Improvement of recall
- **Ensemble impact**
  - ColNet$_{\text{Ensemble}}$ > ColNet
  - Improvement of precision
- **Comparison with the state-of-the-art**
  - ColNet$_{\text{Ensemble}}$ and ColNet > T2K Match
  - ColNet$_{\text{Ensemble}}$ and ColNet has competitive precision as Efthymiou17-Vote, but much higher recall and F1 score

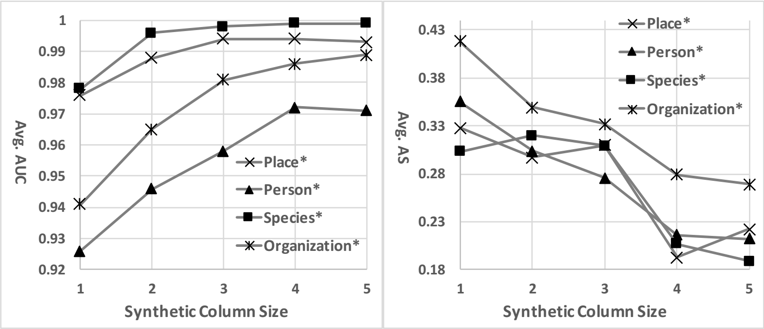
Overall Results on T2Dv2

Precision (P), Recall (R), F1 score (F)

Models	Methods	All Columns			PK Columns		
		P	R	F	P	R	F
Tolerant	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	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

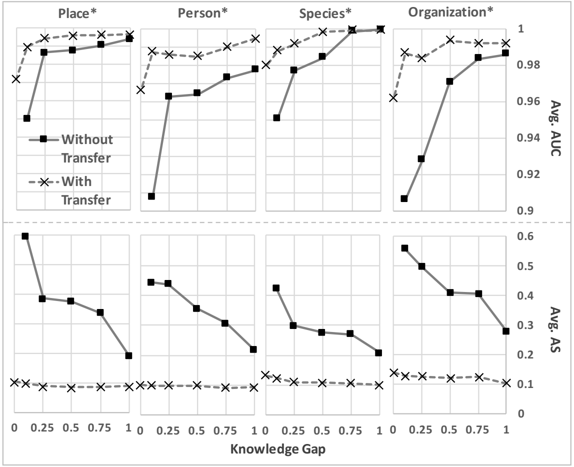
- **Prediction impact**
- **Ensemble impact**
- **Knowledge gap impact**
  - Limaye is harder than T2Dv2
  - Limaye has shorter columns in average, which causes larger knowledge gap
  - Improvements of ColNet$_{\text{Ensemble}}$ and ColNet on Limaye are more significant, since ColNet deals with the knowledge gap

Impact of synthetic column size on CNNs



The testing **performance of CNNs on Truly Matched (TM)** classes [left] and **Falsely Matched (FM)** 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.**

Future Work

Future work

- Learning stronger table locality features (contextual semantics)
- Use ColNet as the basis for other Web table to KG matching tasks
- Application of ColNet in the data science pipeline as an AI assistant
  - Communication with *ptype* and *DataDiff*
- Iterative creation of a shared KG: general knowledge, output from AI assistants, semantic data governance, etc.
- Proposal of a Semantic Web Challenge on Tabular Data to Knowledge Graph Matching

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/`