Entity-based data integration using knowledge bases

Toshiyuki Amagasa

Database Group, CCS, U Tsukuba

amagasa@cs.tsukuba.ac.jp

Our group: KDE
Knowledge & Data Engineering Lab.

Toshiyuki Amagasa, Professor

Accerelators for big data
RDF data mngmnt

Hiroaki Shiokawa, Associate Professor

Graph processing
Graph mining

Hiroyuki Kitagawa, Professor

Stream processing
Data mining

Kazumasa Horie, Assistant Professor

Machine learning
Deep learning

Bou Savong, Assistant Professor

Stream processing and analysis

Yasuhiro Hayase, Assistant Professor

Software engineering
Repository mining

筑波大学 University of Tsukuba

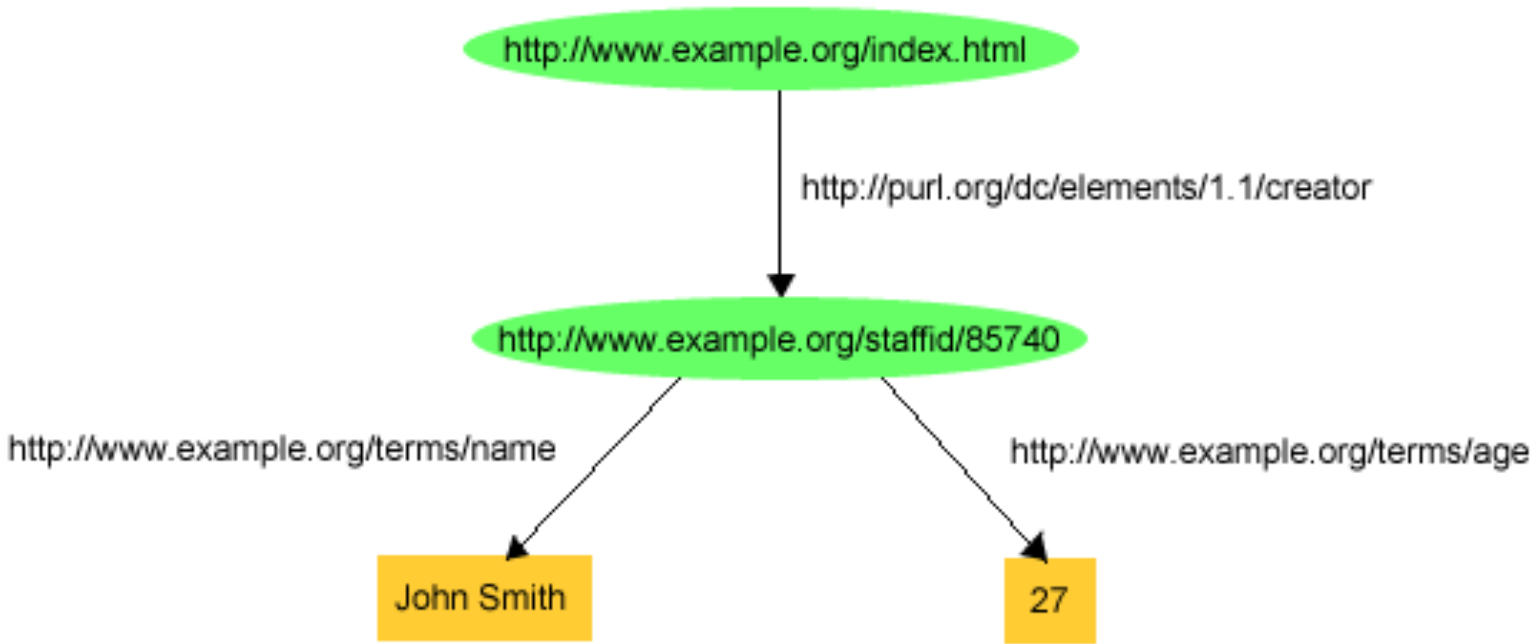
Heterogenous data integration is HARD

- More than 80% of cost (time, money, human, etc.) for data analysis is spent for data integration.
  - AI/ML models need high-quality training data.
  - Making high-quality training data requires huge costs.
- E.g.,
  - Real-life data base schema contains
    - Hundreds of tables with hundreds of attributes.

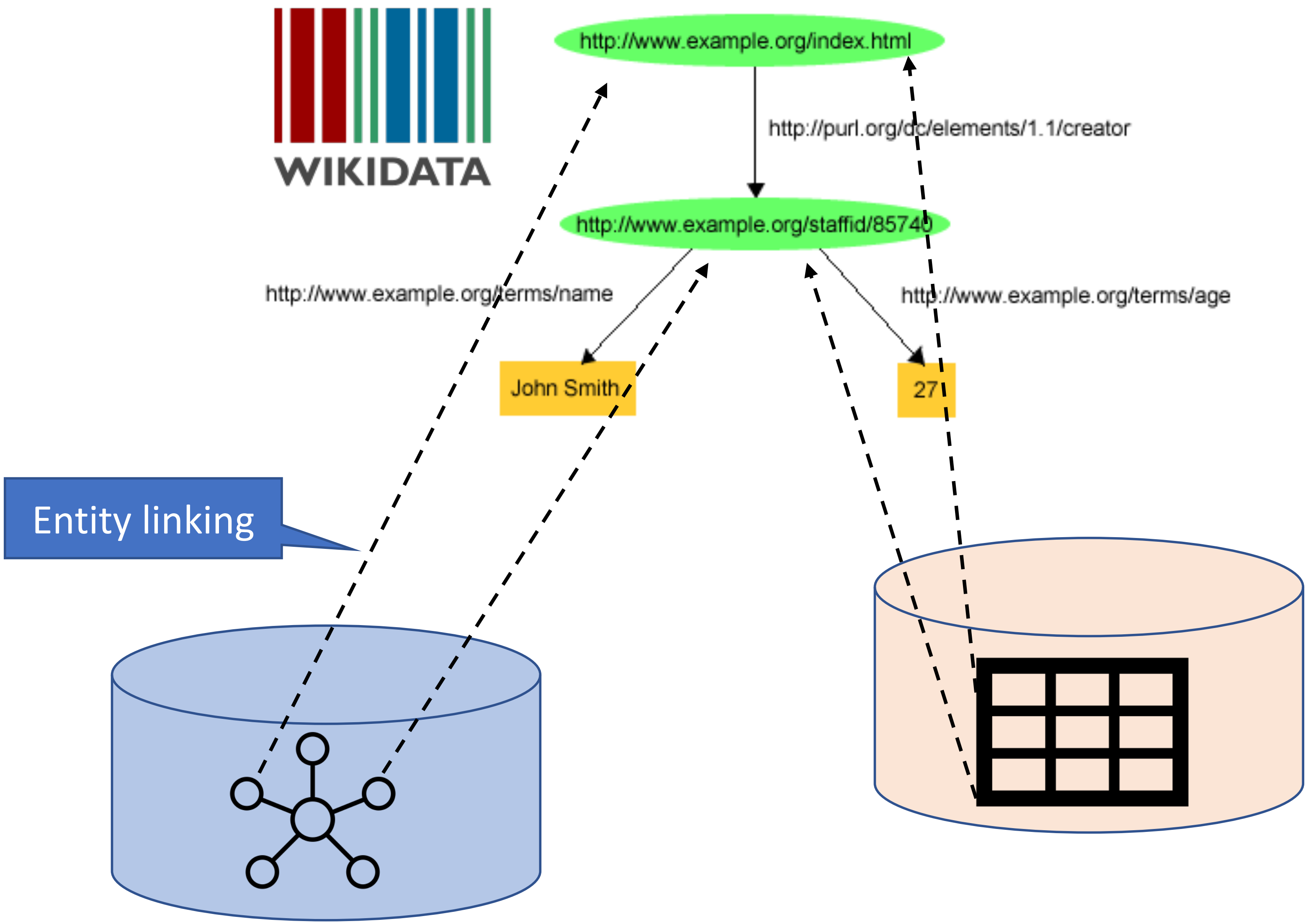
Customer (… , name, …, name2, …, name_new, …)

Knowledge bases

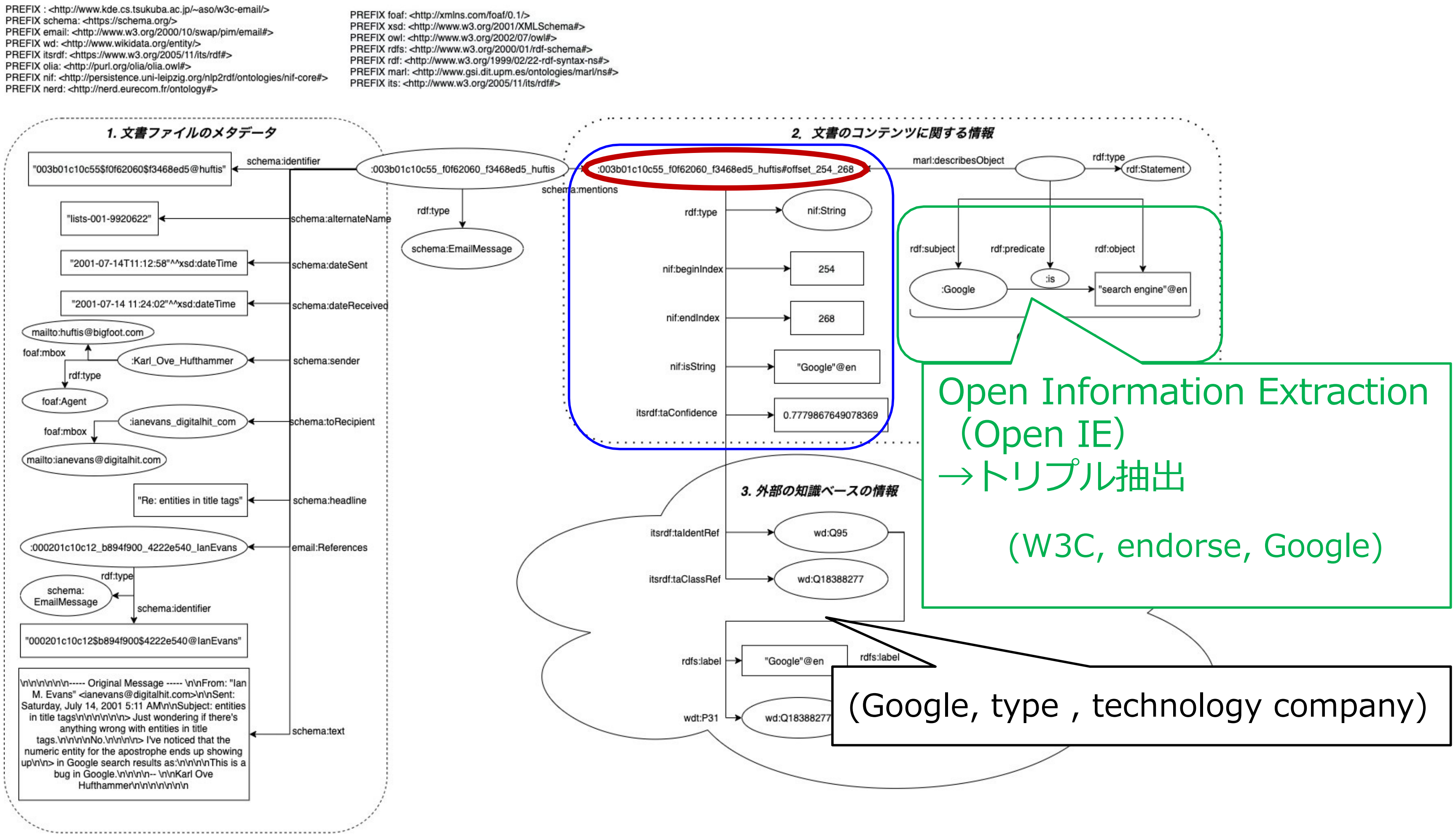
- Large collections of knowledge about real-world entities.
  - Typically modeled as labeled directed graphs.
- Many companies maintain heterogeneous information using KBs.
  - IT companies, drug companies, ..



Entity as a clue for data integration



Entity-based document search [DEIM’21]



Querying multiple KBs and text [iiWAS'21]

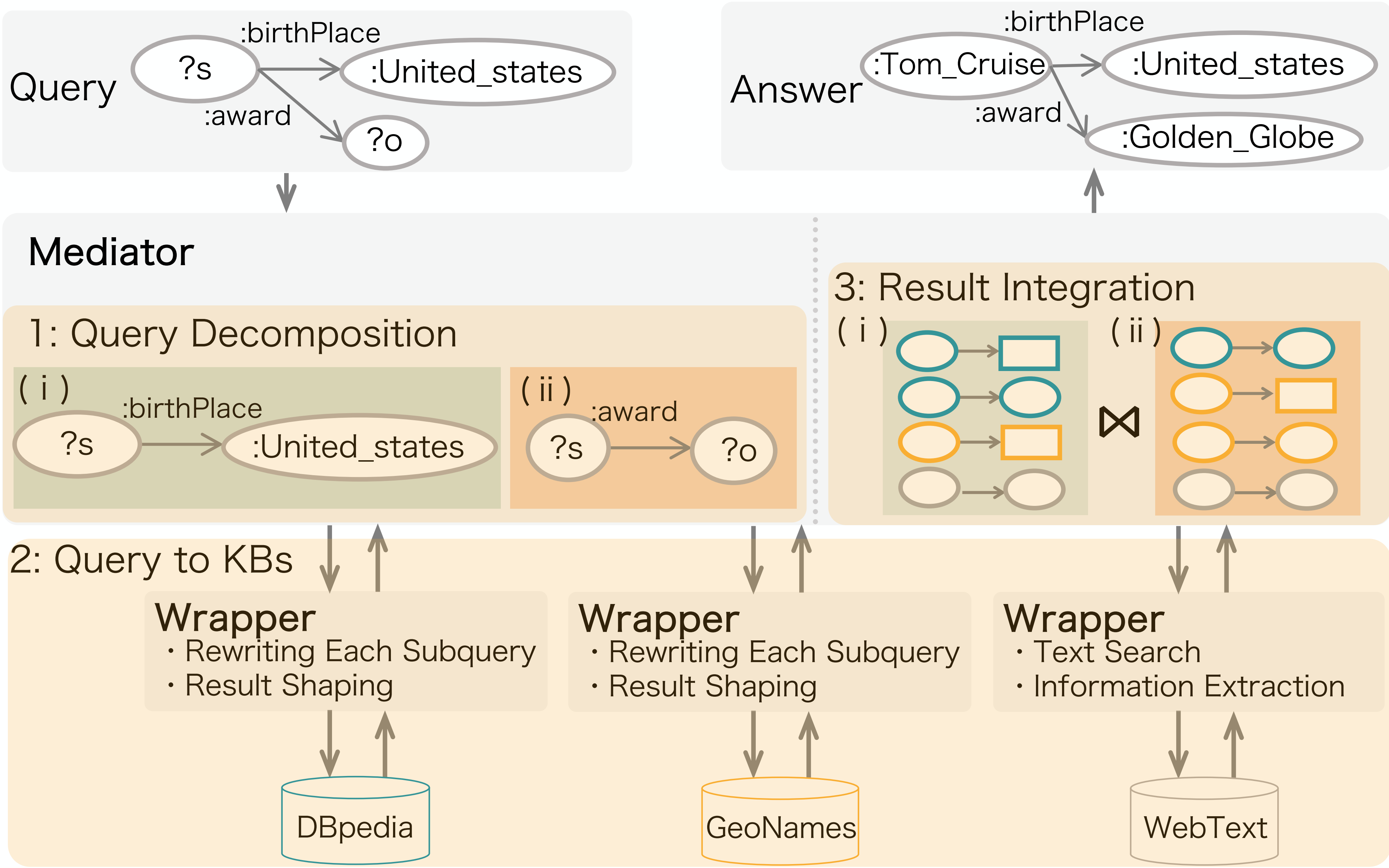
Main Ideas

Distributed querying to multiple KBs using a **mediator/wrapper approach** to deal with heterogeneity in vocabulary and schema.

- The **wrapper** corresponding to each KBs reconstructs and performs SPARQL queries, and the **mediator** integrates the results of each wrapper.
  ↔ Traditional Federated queries require a user to specify the sources and vocabulary.

- We assume a single universal mediated schema. (DBpedia is used in this study.)

Proposed method | Framework



Evaluation | Overview

Purpose

Evaluate the improvement of coverage by rewriting to multiple KBs and text information resources.

KBs

- DBPedia
- GeoNames

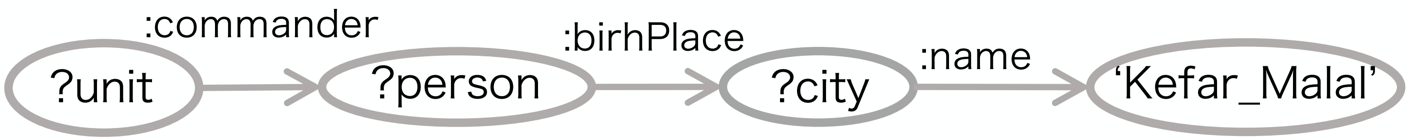
Text Information Sources

- Reverb45K
  (With 36,000 sentences extracted from news text)

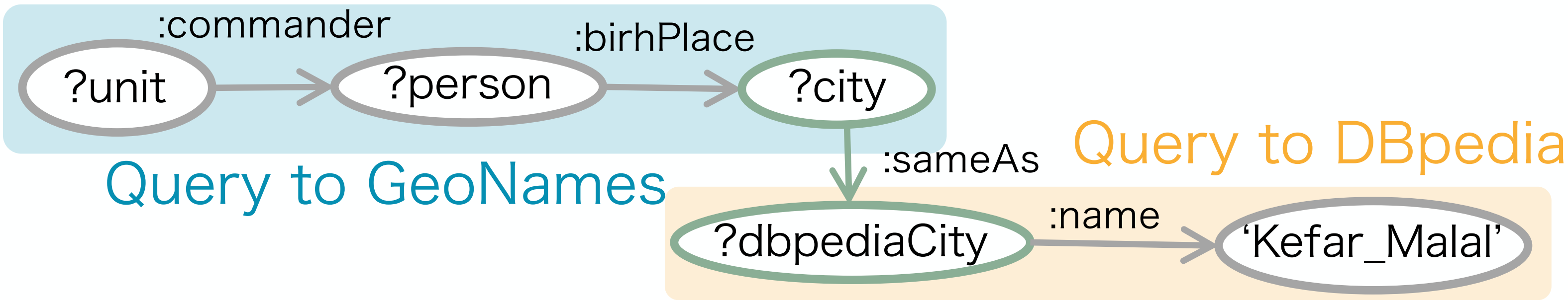
Evaluation | Queries

- Created 10 transversal queries to DBPedia and GeoNames based on Fed-bench, federated query benchmark.

(a) Query assuming mediated schema

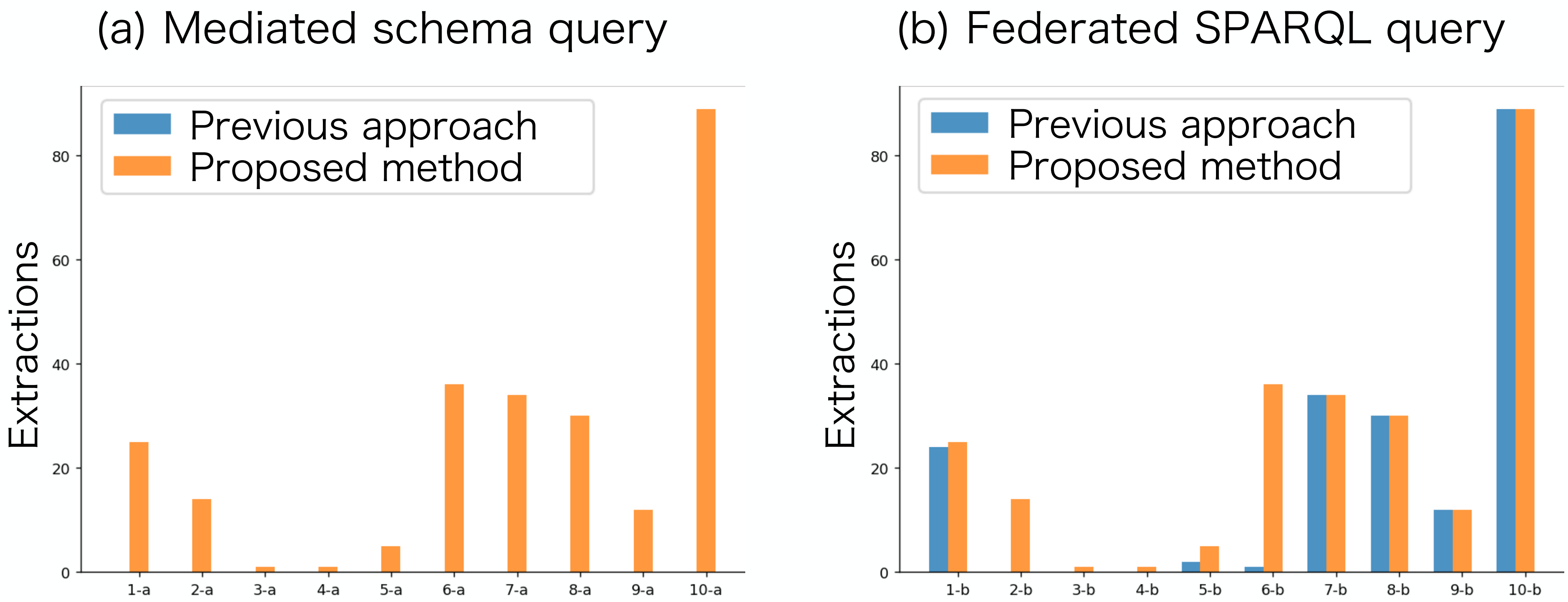


(b) Federated SPARQL query



Evaluation | Number of results retrieved

- Query results for each of the 20 queries



Allows users to perform federated queries without considering the heterogeneity of schemas between KBs.

A Graph-based Blocking Approach for Entity Matching Using Contrastively Learned Embeddings

Presenter: **John Bosco Mugeni**
Supervisor: Toshiyuki Amagasa
University of Tsukuba

Published in (ACM SIGAPP) Applied Computing Review, 2022
Computer Science

Introduction: entity matching

- Entity Matching: is the task of discovering matching entries among disperate data sources.
- The goal is to then link these entries with a high-match quality
- However, the process meets quadratic complexity problem w.r.t dataset size

Table A

category	brand	model no.	price
garden - general	d-link	dcs-1100	99.82
furniture	3m	fr530cb	67.88
stationery & office machinery	brother	dk2113	64.88

Table B

category	brand	model no.	price
footrests	3m #	fr530cb #	67.34
file folder labels	avery	5029	14.2
surveillance cameras	d-link	dcs-1100	99.82

Figure: An example of matching tuples

Introduction: blocking

- "Blocking" is introduced for efficient execution of entity matching
- The naive pairwise comparison (right figure) requires exorbitant computation due to a massive search space in contrast to a partitioned search space due to "blocking" (left figure)

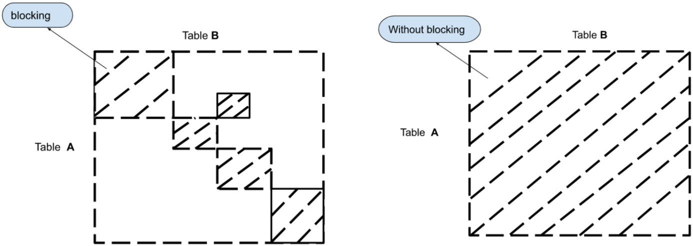


Figure: Types of blocking frameworks

Introduction: blocking techqniues

- "Blocking" techniques can be categorized into 3 types;

Figure: Types of blocking frameworks

- Rule-based methods require handcrafted features, domain knowledge & are labour intensive
- Learning-based methods have high accuracy but require labelled data (labels are not always available)
- Cluster-based methods circumvent the need of labels & handcrafted features

Thesis objective and contributions

- We propose a graph-based blocking technique predicated on the k-nearest neighbour (k-NN) graph algorithm for EM.
- We leverage readily available context-aware sentence embeddings from four pre-trained language models for our blocking scheme
- We show that our k-NN graph blocking transcends the existing deep learning-based cluster blocking solution in terms of time and accuracy.

Related works

- Later the paper of Azzalini[1] develops a system for “blocking” based on the RNN architecture.
    - However, clustering large data sets proves to be resource-intensive
    - Morever, vectors have to be down-sampled via the t-SNE algorithm, in their work, which scales poorly on big data sets
    - The RNN architecture relies on simple word embeddings that neglect context

[1] F Azzalini, et al. 2020. Blocking Techniques for Entity Linkage: A Semantics-Based Approach.

Proposed approach: system overview

An overview of the system is as follows;

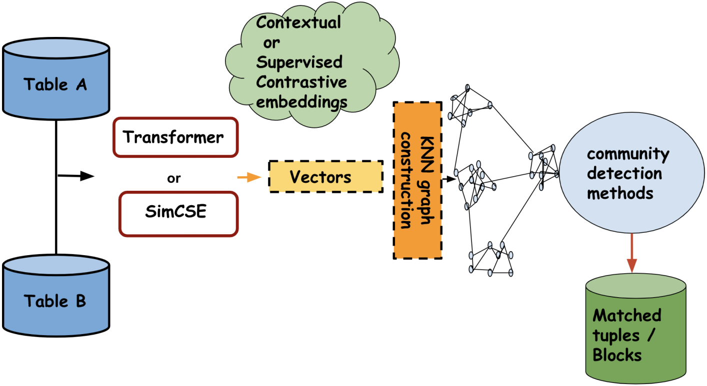


Figure: Our blocking system

Proposed approach: pipeline step 1

First, attributes of data sets to be integrated are concatenated into a string

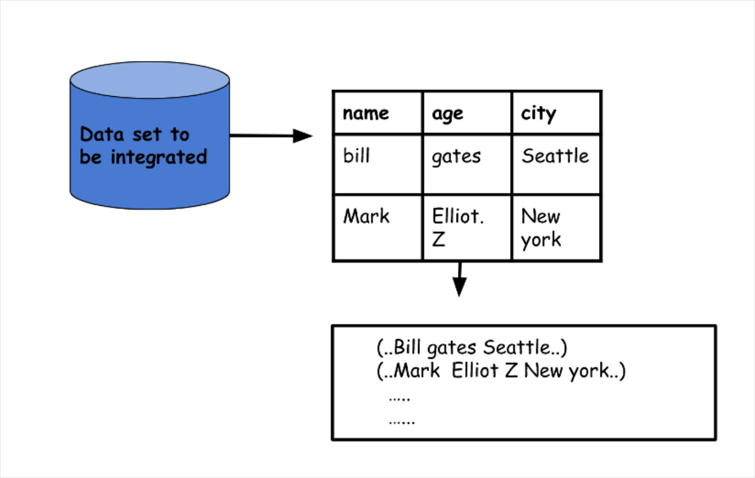


Figure: Textual representation from table A or B

Proposed approach:pipeline step 2

Next, each tuple is then input to a pre-trained transformer language model producing context embeddings

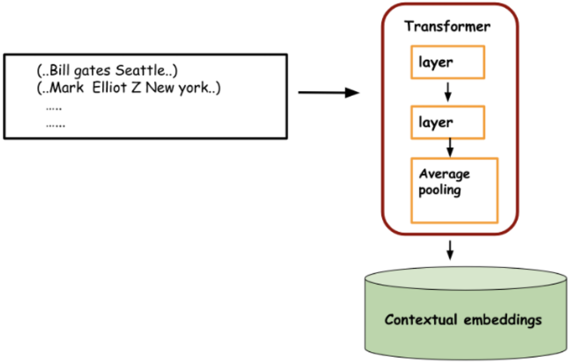


Figure: Feature extraction (generating embeddings)

Proposed approach:pipeline step 3

Projection of embeddings to lower dimension is possible via UMAP or CVAE

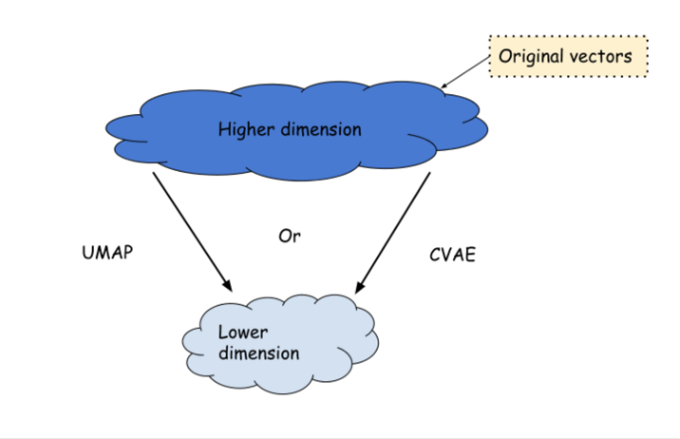


Figure: elaborating the vector processing in case of dimensionality reduction

Proposed approach: pipeline step 4

Next, we apply knn graph algorithm on embedding vectors to construct a graph followed by unsupervised community detection algorithms

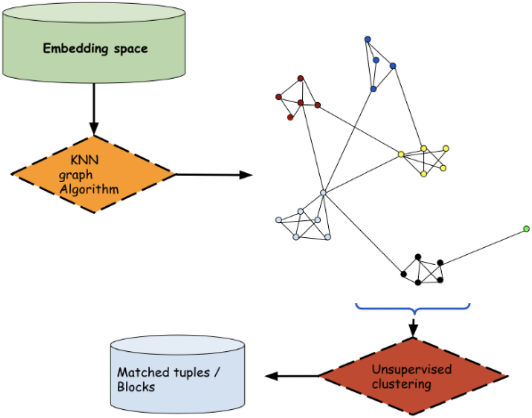


Figure: KNN-graph based blocking

Experimental work: data sets

- Each data set has the format of **Table A-Table B**
- Each pair has more than 6 million record comparisons

Table 5: Dataset statistics.

Data	Domain	#Tuples	#Matches	Attr	Size (M)
DBLP-Scholar,	citation	2616-64263	5347	4	168
iTunes-Amazon	music	6907-55923	132	8	386
Walmart-Amazon	electronics	2554-22074	962	5	56
GoogleScholar-DBLP	citation	2616-64263	5347	4	168

Figure: Experimental datasets for entity matching

Experimental work: computing environment & key parameters

- For the transformer based models, we choose the attention spans to be 200 tokens
- Batch size is chosen to be 32 & mean-pooling for summarising input tokens
- A single workstation equipped with Intel(R) Core(TM) i7-4820K quad-core CPU encompassing 48 GB RAM running Ubuntu 18.04
- We use pre-trained models based on Hugging-face [2] & all programs are executed in python version 3.7.6

[2] T. Wolf et al. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:cs.CL/1910.03771

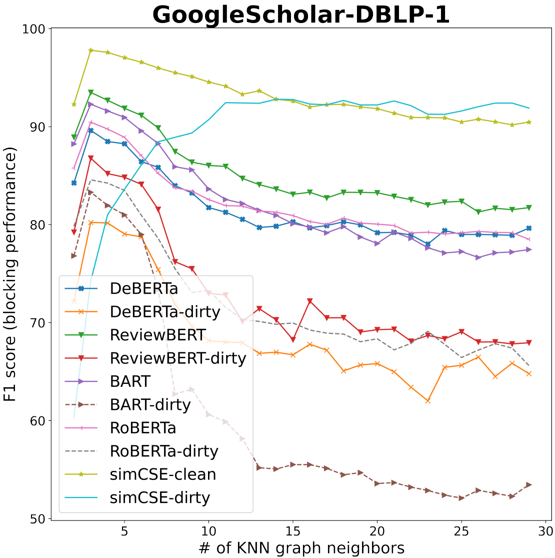
Results: blocking time

Table 5: iTunes-Amazon.

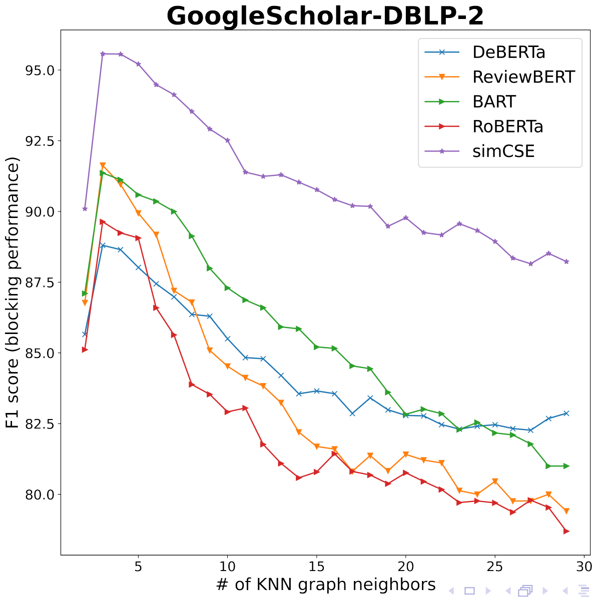
method	$algo_{best}$	emb'_{sec}	bk_{sec}	$total_{sec}$	F1
R-BERT	l'vian	**91.8**	461.8	553.6	85.2
DeBERTa	l'vian	311.2	557.2	868.4	89.2
RoBERTa	l'vian	253.6	**58.1**	**311.7**	89.7
BART	l'vian	324.0	433.6	757.6	**91.7**
RNN	birch	2329.8	dnf	dnf	dnf
SimCSE	l'vian	64.5	160.9	225.4	92.8
$R\text{-}BERT_d$	l'den	127.7	**328.2**	**455.9**	56.2
$DeBERTa_d$	l'den	470.0	607.8	1077.8	56.4
$RoBERTa_d$	l'vian	391.5	368.2	759.7	64.0
$BART_d$	l'den	642.9	347.5	990.4	**68.0**
$SimCSE_d$	l'den	125.8	164.4	290.2	89.7

Figure: Performance on iTunes-Amazon(62,830 tuples)

Comparison of embeddings as a function of parameter k



Comparison of embeddings as a function of parameter k



Conclusion

- As future work, we plan to improve representation learning using task domain data as well combining our approach with a supervised system for Entity Matching.