Étude de transférabilité des clés pour le liage de données entre graphes de connaissances

Thibaut Soulard, Fatiha Saïs, Joe Raad, Gianluca Quercini
LISN, CNRS (UMR 9015), Université Paris Saclay, France

LABORATOIRE INTERDISCIPLINAIRE DES SCIENCES DU NUMÉRIQUE

Ingénierie des Connaissances
04 juillet 2023, Strasbourg, France

Motivation : linking entities across knowledge graphs

DBpedia

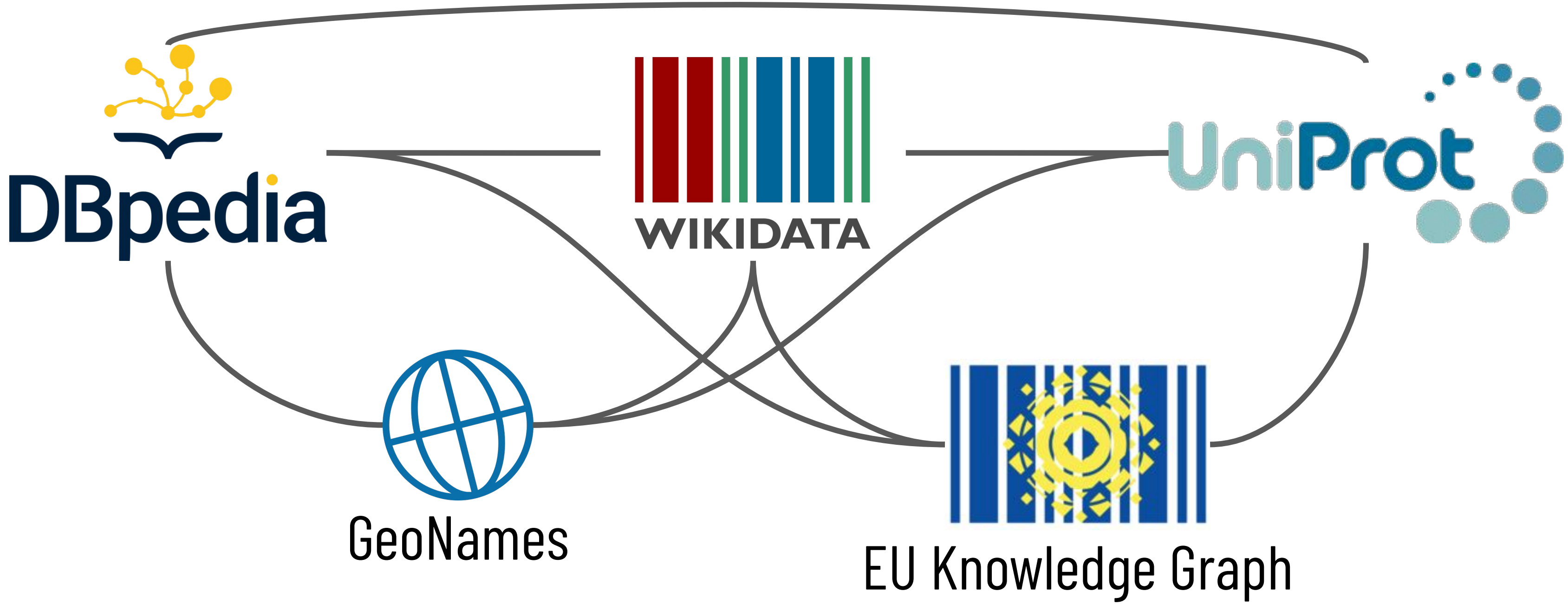
WIKIDATA

UniProt

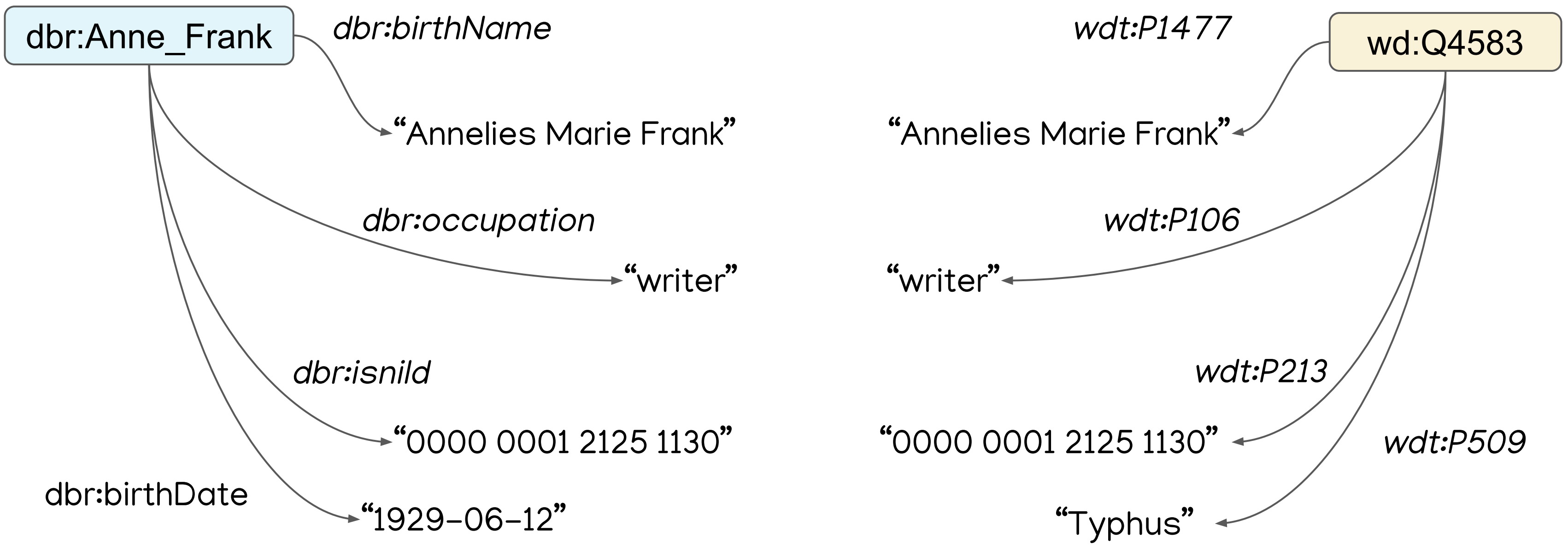
GeoNames

EU Knowledge Graph

Motivation : linking entities across knowledge graphs



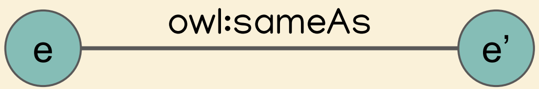
Entity linking problem in heterogeneous KGs



Motivation : linking entities across knowledge graphs

Machine learning based

- Embeddings, Tensors, GNN, ...
- Requires seeds



- Harder to explain to neophytes

Key based

- Does not require seeds
- Easy to explain
- Time consuming

Definition of Keys and Exception rate

We consider S-Keys [Atencia et al 14] that follows the semantics of OWL2 keys (**owl:hasKey**)

$$\forall x \forall y \forall z_1 ... z_n (C(x) \wedge C(y) \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow \ x = y)$$

$$K = \{dbr : birthName, dbr : occupation\}$$

	BirthName	***Occupation***
e_1	Annelies Marie Frank	Writer
e_2	Alan Mathison Turing	
e_3	Alice Allison Dunnigan	Writer
e_4	Michelle Williams	Actor Singer
e_5	Michelle Williams	Actor TV host

[Atencia et al 14] Defining Key Semantics for the RDF Datasets: Experiments and Evaluations. ICCS 2014: 65-78

Definition of Keys and Exception rate

We consider S-Keys [Atencia et al 14] that follows the semantics of OWL2 keys (**owl:hasKey**)

$$\forall x \forall y \forall z_1 ... z_n (C(x) \wedge C(y) \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow \ x = y)$$

$$K = \{dbr : birthName, dbr : occupation\}$$

	BirthName	***Occupation***
e_1	Annelies Marie Frank	Writer
e_2	Alan Mathison Turing	
e_3	Alice Allison Dunnigan	Writer
e_4	Michelle Williams	Actor Singer
e_5	Michelle Williams	Actor TV host

[Atencia et al 14] Defining Key Semantics for the RDF Datasets: Experiments and Evaluations. ICCS 2014: 65-78

Definition of Keys and Exception rate

We consider S-Keys [Atencia et al 14] that follows the semantics of OWL2 keys (**owl:hasKey**)

$$\forall x \forall y \forall z_1 ... z_n (C(x) \wedge C(y) \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow x = y)$$

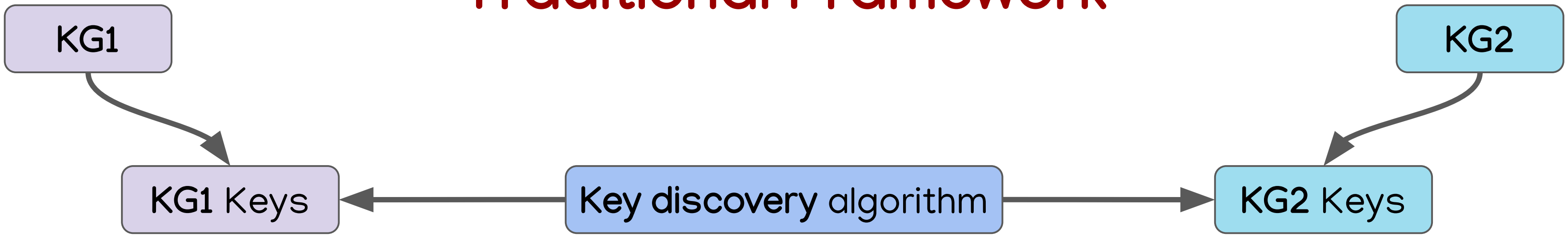
$$K = \{dbr : birthName, dbr : occupation\}$$

$$ex(k) = 2 \qquad ex(k)_{norm} = \frac{ex(k)}{\#Entities} = \frac{2}{5}$$

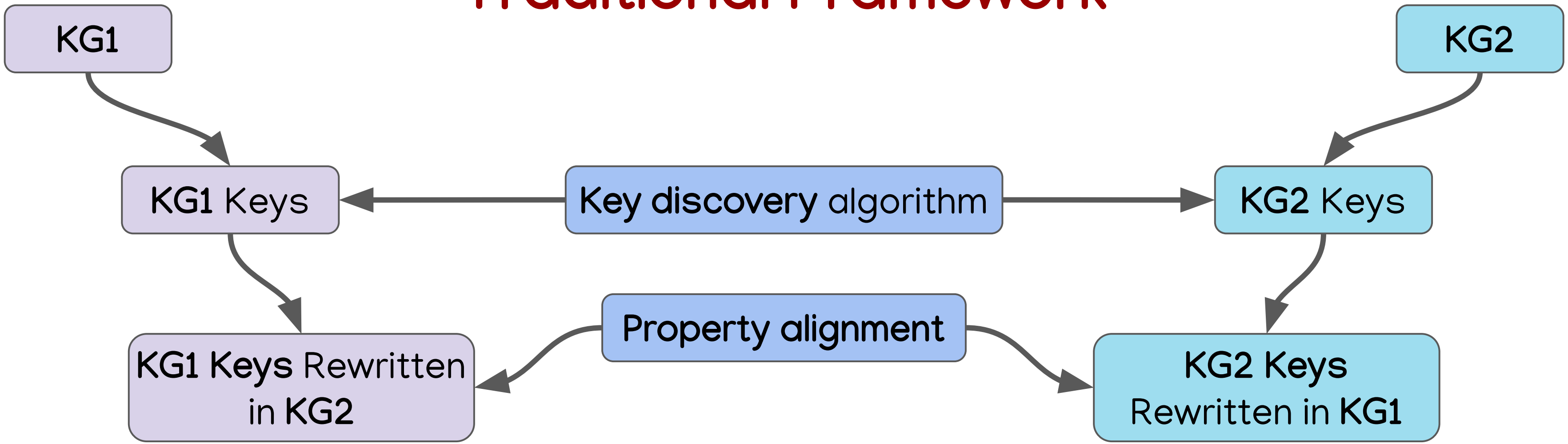
	BirthName	***Occupation***
e_1	Annelies Marie Frank	Writer
e_2	Alan Mathison Turing	
e_3	Alice Allison Dunnigan	Writer
e_4	Michelle Williams	Actor Singer
e_5	Michelle Williams	Actor TV host

[Atencia et al 14] Defining Key Semantics for the RDF Datasets: Experiments and Evaluations. ICCS 2014: 65-78

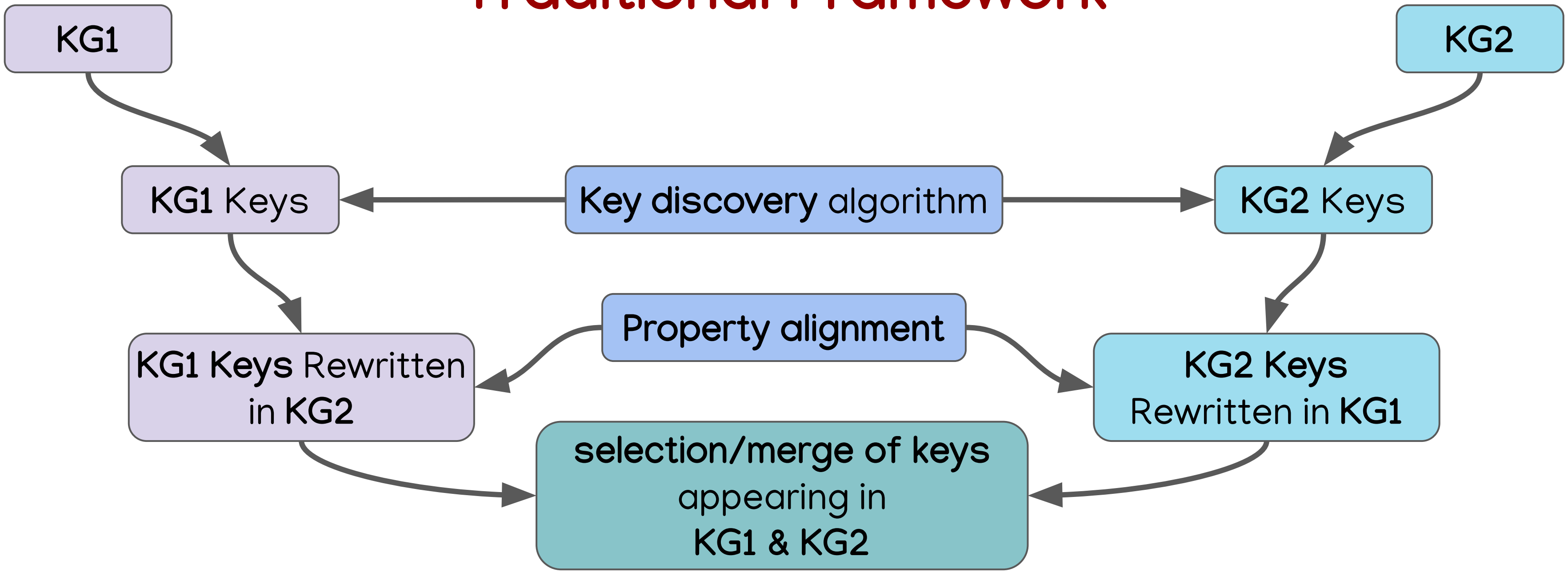
Traditional Framework



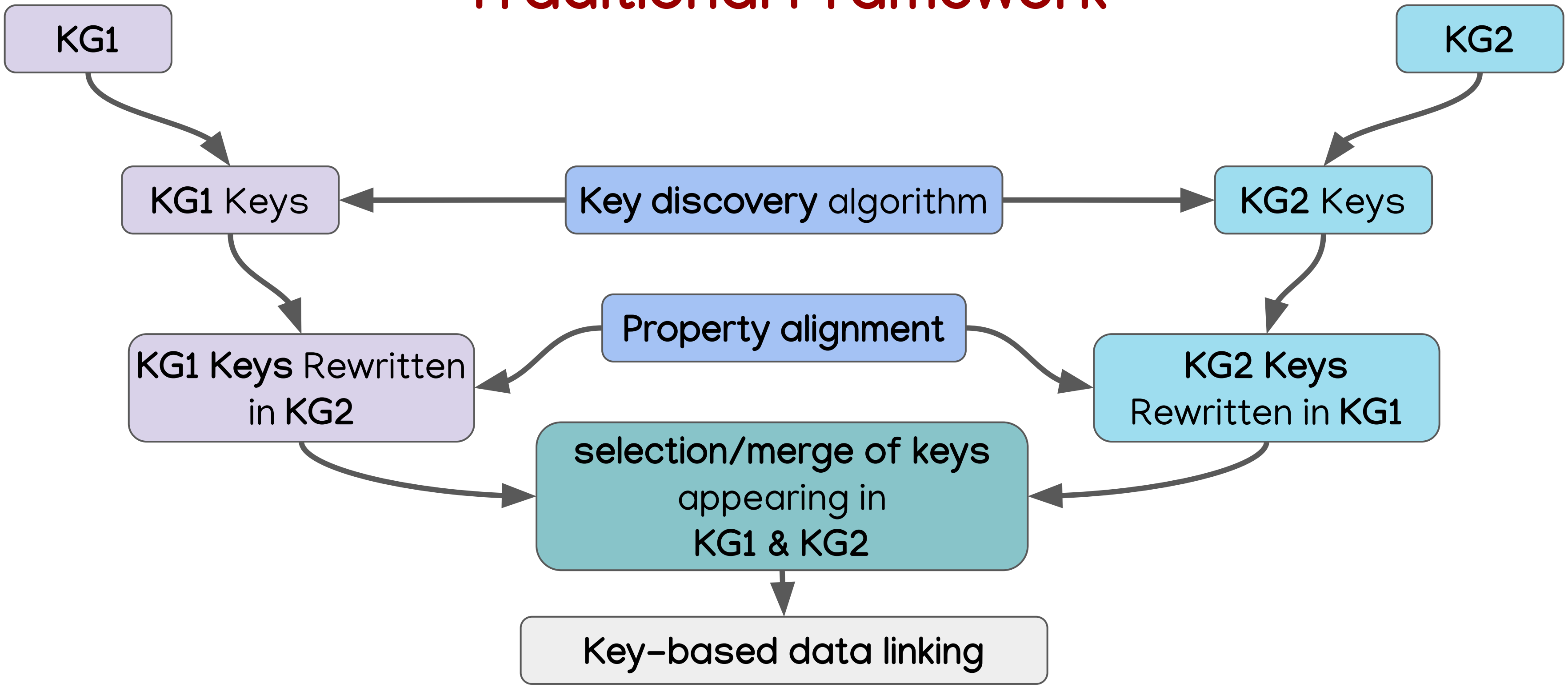
Traditional Framework



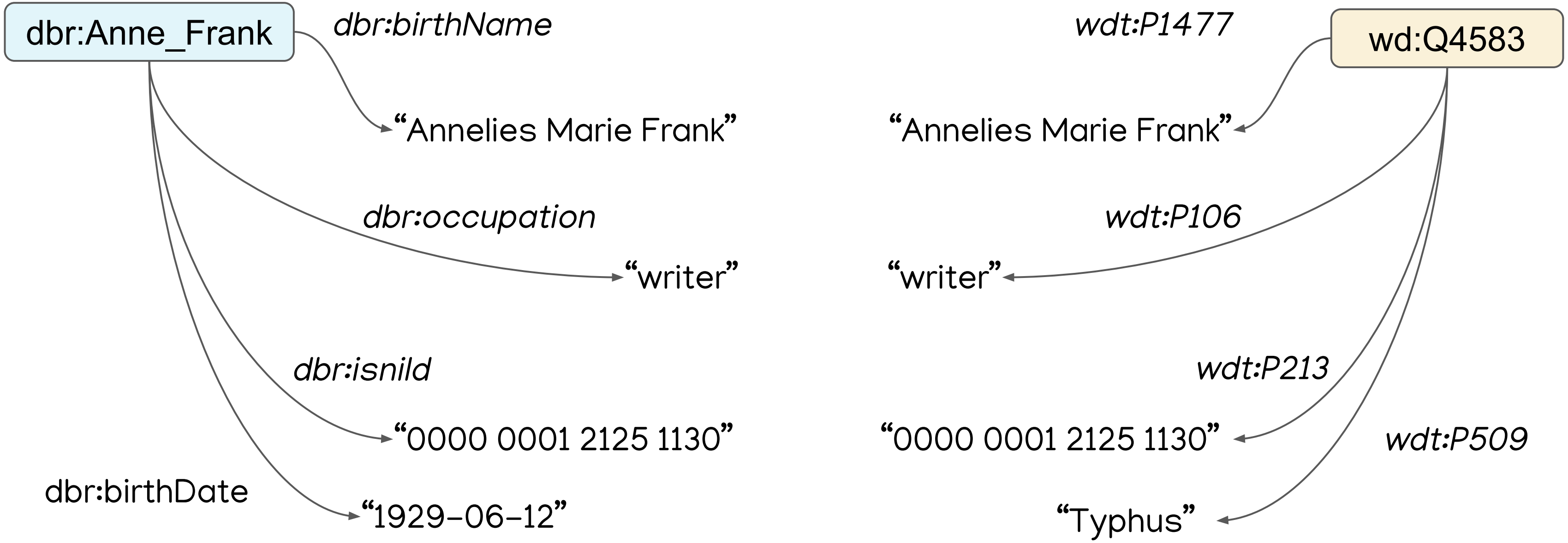
Traditional Framework



Traditional Framework



Running example of a key-based entity linking



Running example of a key-based entity linking

Key discovery algorithm

$$K_1 = \{dbr : isnId\}$$

$$K'_1 = \{wdt : P1477, wdt : P106\}$$

$$K'_2 = \{wdt : P213\}$$

$$K'_3 = \{wdt : P1477, wdt : P509\}$$

Running example of a key-based entity linking

Key discovery algorithm

Property alignment

dbr:birthName · *wdt:P1477*

EquivalentProperty

dbr:occupation · *wdt:P106*

EquivalentProperty

dbr:isnild · *wdt:P213*

EquivalentProperty

dbr:birthDate *wdt:P509*

Running example of a key-based entity linking

Key discovery algorithm

Property alignment

Key rewriting

dbr:birthName ············ wdt:P1477

$$K'r_1 = \{(wdt : P1477 \equiv dbr : birthName), (wdt : P106 \equiv dbr : occupation)\}$$

dbr:occupation ············ wdt:P106

dbr:isnild ············ wdt:P213

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

$$K'r_2 = \{(wdt : P213 \equiv dbr : isnId)\}$$

Running example of a key-based entity linking

Key discovery algorithm

Property alignment

Key rewriting

$$K'r_1 = \{(wdt : P1477 \equiv dbr : birthName), (wdt : P106 \equiv dbr : occupation)\}$$

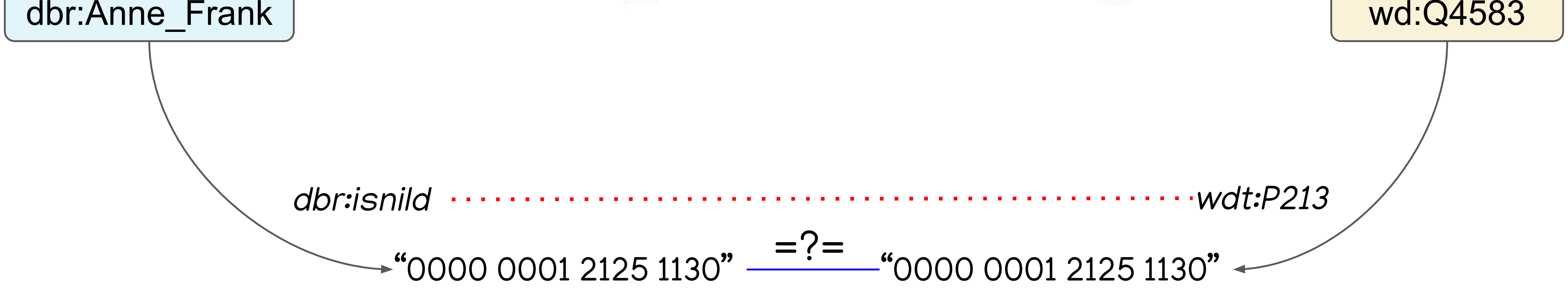
Key merge

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

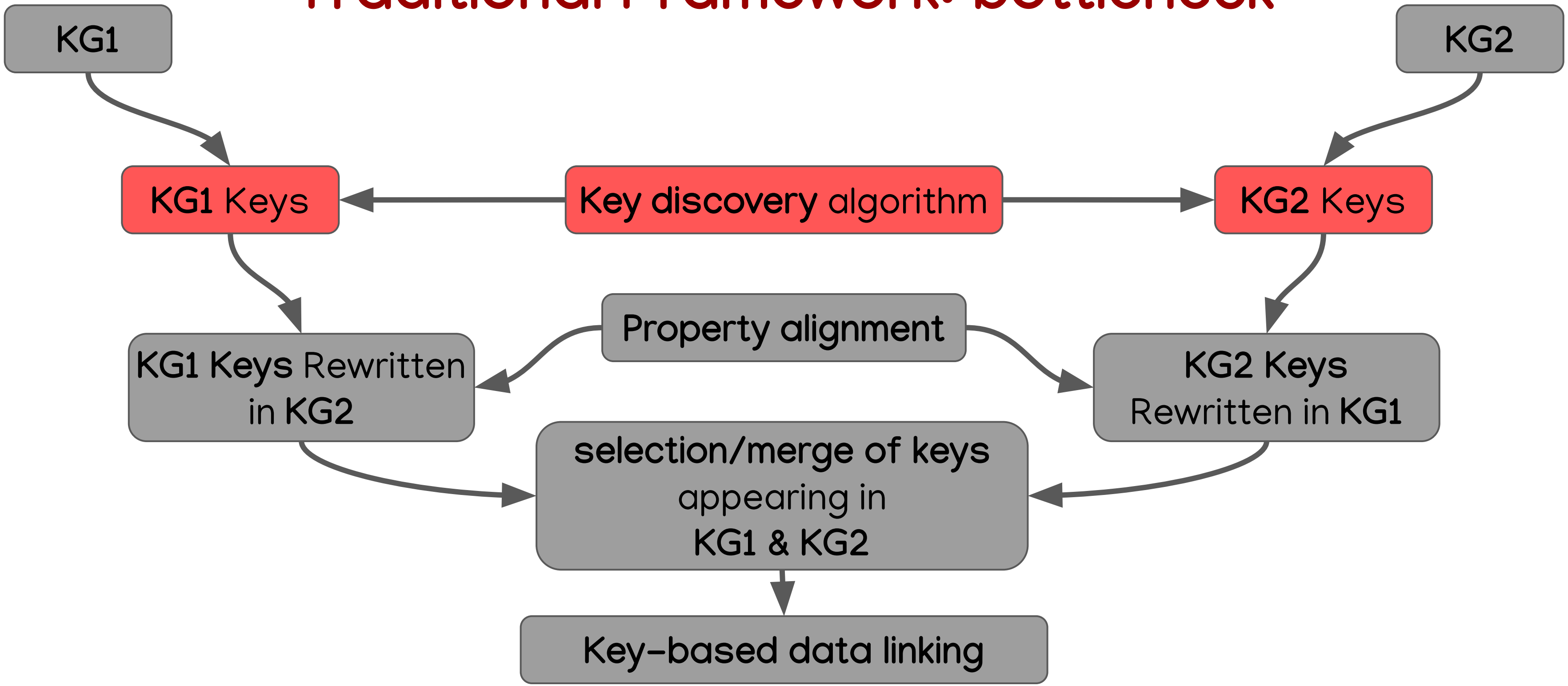
$$K'r_2 = \{(wdt : P213 \equiv dbr : isnId)\}$$

Running example of a key-based entity linking

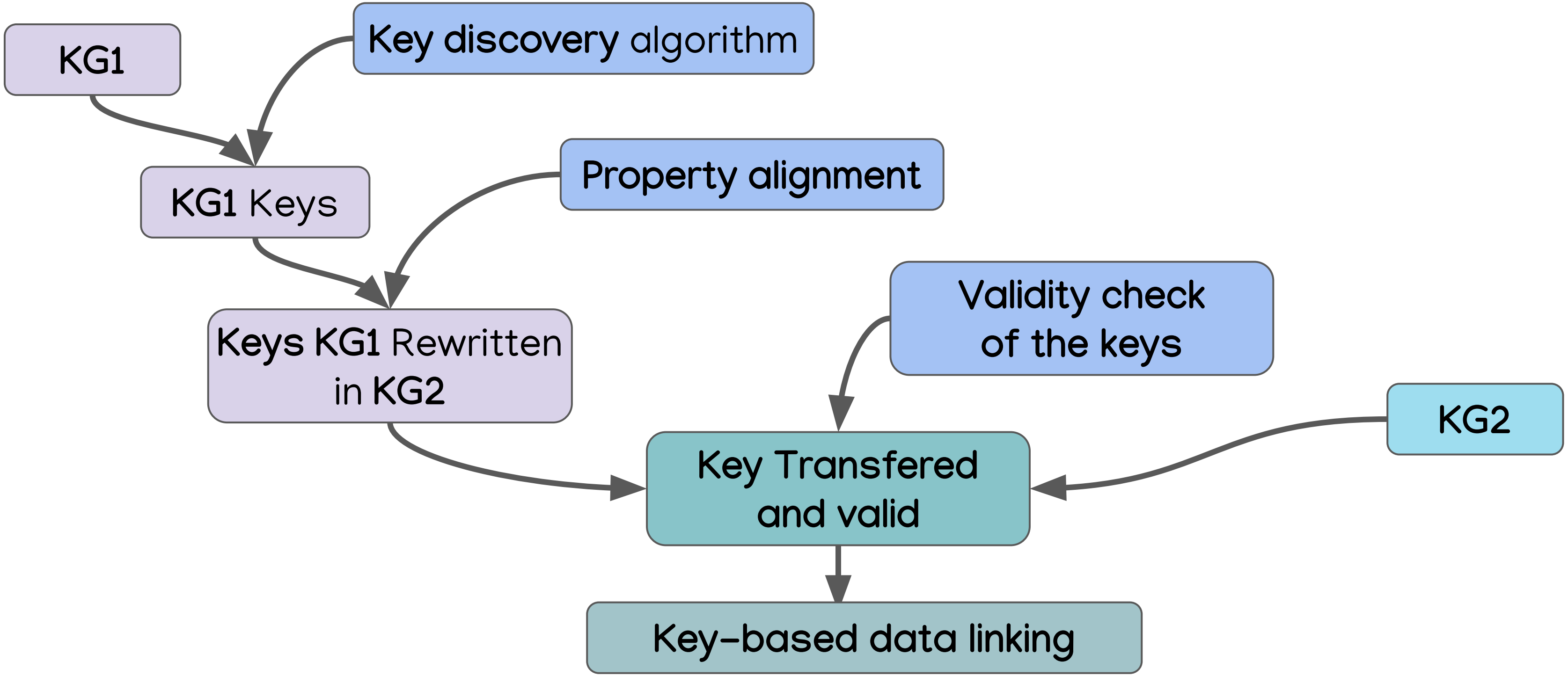
$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$



Traditional Framework: bottleneck



Key transfer-based framework



Validity check of the keys

Is a key found in KG1 and rewritten in KG2 be a key in the KG2 ?

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

Validity check of the keys

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

	P1477 (BirthName)	*P106 (Occupation)*	*P213 (isnild)*
e_1	Annelies Marie Frank	Writer	0000 0001 2125 1130
e_2	Alan Mathison Turing		
e_3	Alice Allison Dunnigan	Writer	0000 0000 2348 3667
e_4	Michelle Williams	Actor Singer	
e_5	Michelle Williams	Actor TV host	

Validity check of the keys

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

	P1477 (BirthName)	*P106 (Occupation)*	*P213 (isnild)*
e_1	Annelies Marie Frank	Writer	0000 0001 2125 1130
e_2	Alan Mathison Turing		
e_3	Alice Allison Dunnigan	Writer	0000 0000 2348 3667
e_4	Michelle Williams	Actor Singer	
e_5	Michelle Williams	Actor TV host	

$$ex(Kr_2)_{norm} = 0$$

Validity check of the keys

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

	P1477 (BirthName)	P106 (Occupation)	P213 (isnild)
e_1	Annelies Marie Frank	Writer	0000 0001 2125 1130
e_2	Alan Mathison Turing		
e_3	Alice Allison Dunnigan	Writer	0000 0001 2125 1130
e_4	Michelle Williams	Actor Singer	
e_5	Michelle Williams	Actor TV host	

$$ex(Kr_2)_{norm} = 0.2$$

Validity check of the keys

$$Kr_1 = \{(dbr : isnId \equiv wdt : P213)\}$$

	P1477 (BirthName)	P106 (Occupation)	P213 (isnId)
e_1	Annelies Marie Frank	Writer	0000 0001 2125 1130
e_2	Alan Mathison Turing		
e_3	Alice Allison Dunnigan	Writer	0000 0001 2125 1130
e_4	Michelle Williams	Actor Singer	
e_5	Michelle Williams	Actor TV host	

$$ex(Kr_2)_{norm} = 1$$

Validity check of the keys

Relative exception rate :

$$ex^R(k) = \frac{ex(k)}{support(k)}$$

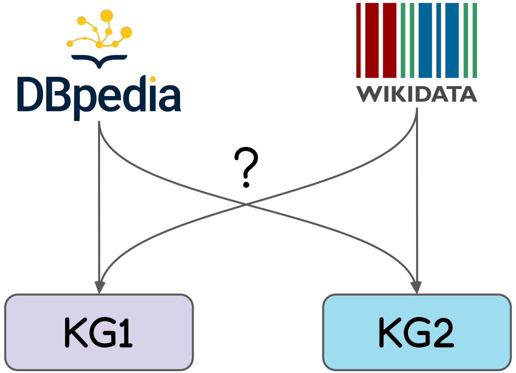
Support : $support(k) = \mid \{x \mid \forall p \in prop(k), \exists y, (x, p, y) \in G\} \mid$

Research questions

- How much time do we gain ?

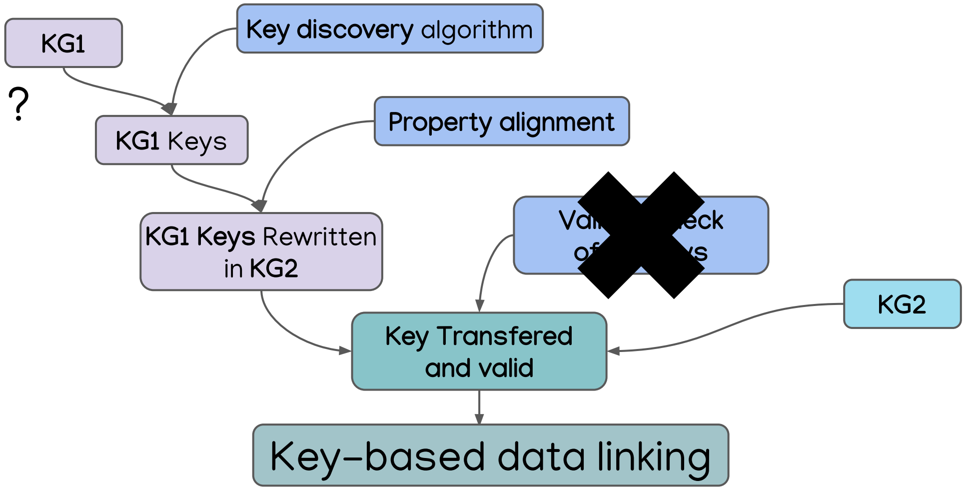
Research questions

- How much time do we gain ?
- Which graph for KG1 and KG2 ?



Research questions

- How much time do we gain ?
- Which graph for KG1 and KG2 ?
- Is the validity check necessary ?



Research questions

- How much time do we gain ?
- Which graph for KG1 and KG2 ?
- Is the validity check necessary ?
- How well this framework can perform on the linkage problem ?

Experiments: datasets

	# Triples	# Entities	# Relations
Wikidata : Q5	7 503 002	3 020 916	135
DBpedia : Human	12 474 844	1 863 013	239

Knowledge Graphs stats

Evaluation Results

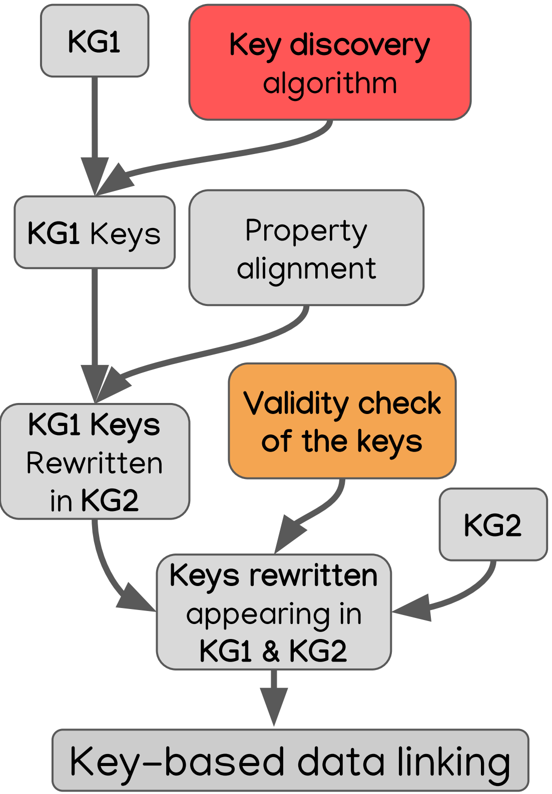
How much time was gained ?

	0	0.5%	1%	2%	3%	4%	5%	10%
New Start DB	100.95%	101.64%	101.61%	101.67%	101.70%	101.69%	101.71%	101.59%
New Start WK	4.52%	4.05%	4.05%	4.19%	4.10%	4.11%	4.08%	4.01%

Time relative of the new framework compared to the typical Framework

With a **good starting point,** we are able to drastically **reduce** the **total running time.**

Evaluation Results



Which graph for KG1 and KG2 ?

	# Triples	# Entities	# Relations
Wikidata : Q5	7 503 002	**3 020 916**	**135**
DBpedia : Human	12 474 844	**1 863 013**	**239**

Knowledge Graphs stats

KG1 should be the graph with the **fewest relations** to reduce the running time.

If we have a **similar number of relations,** ones should prioritize the **fewest entities** for KG2.

Evaluation Results

Is the validity check necessary ?

	ex^{R}_{max}	0%	0.5%	1%	2%	3%	4%	5%	10%
Start DB	**Key Rewritten Not verified**	69	64	64	64	65	65	65	68
	Key Rewritten Verified	49	52	52	54	56	56	56	60
Start WK	**Key Rewritten Not verified**	82	82	82	82	83	83	83	83
	Key Rewritten Verified	41	48	48	49	52	54	55	57

Number of Keys Rewritten before and after Verification

We cannot remove the **validity check** as we have a **great number of keys** that **degenerate.**

Evaluation Results

How well this framework can perform on the linkage ?

	ex^{R}_{max}	0%	0.5%	1%	2%	3%	4%	5%	10%
Start DB	**DB**	0.32%	80.72%	80.72%	80.72%	80.72%	80.72%	80.72%	80.72%
	WK	0.90%	28.88%	28.88%	28.89%	28.89%	28.89%	28.89%	28.89%
Start WK	**WK**	0.31%	0.88%	0.88%	0.88%	1.01%	1.03%	1.03%	1.03%
	DB	0.10%	80.73%	80.73%	80.73%	80.73%	80.73%	80.73%	80.73%

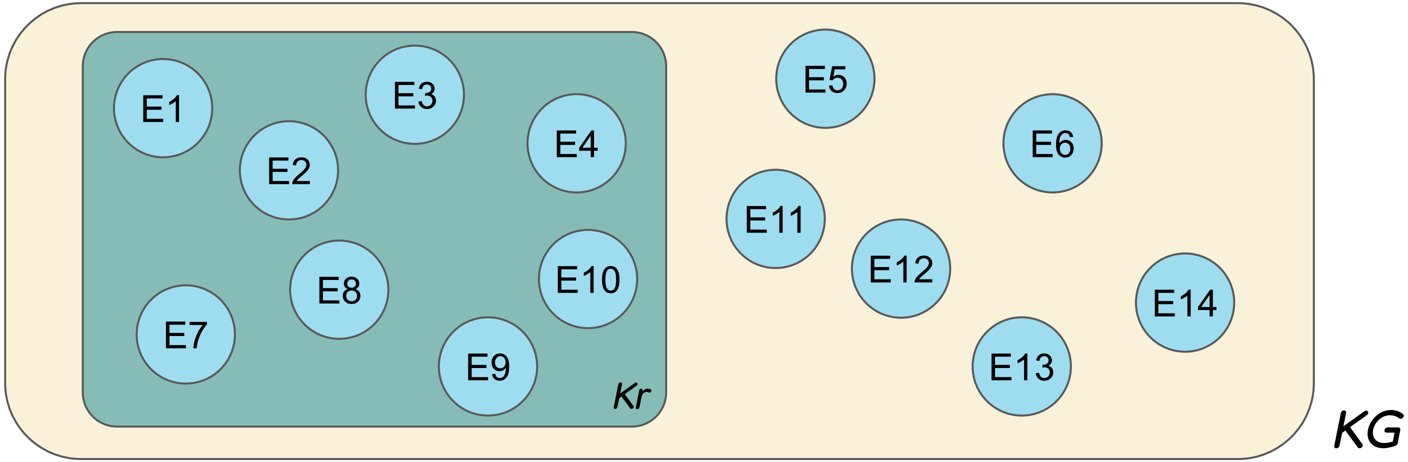
Percentage of entities that are distinguishable from every other by at least a key rewritten

DBpedia performs **well** with this **framework,** while the **start** with **Wikidata** is **poor.**

Evaluation Results

How well this framework can perform on the linkage ?

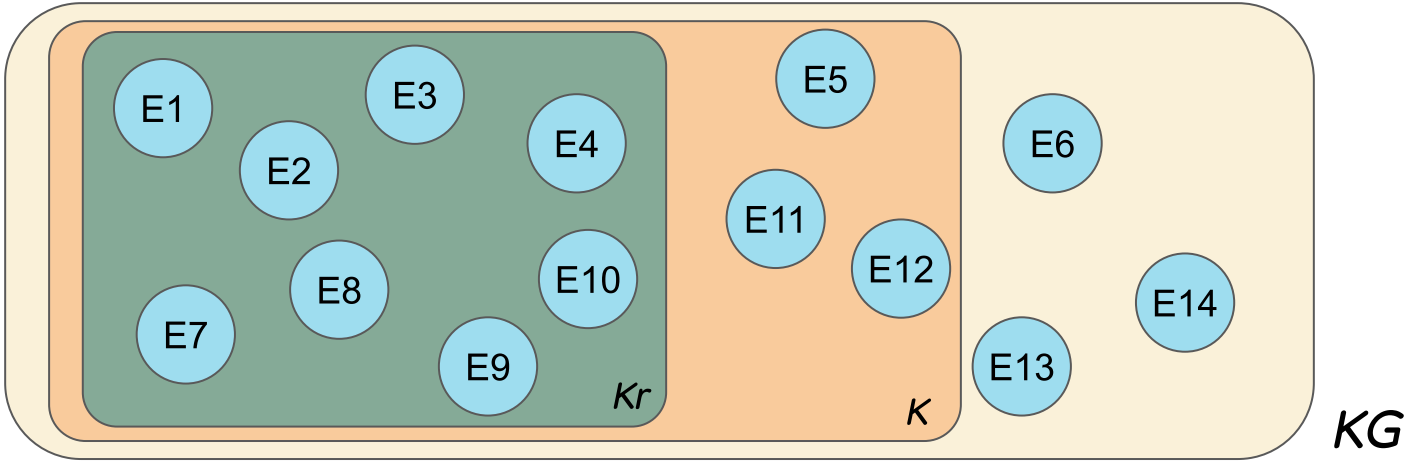
$$\frac{\#\ \textit{Entities Differentiated by a Kr}}{\#\ \textit{Entities}}$$



Evaluation Results

How well this framework can perform on the linkage ?

$$\frac{\#\ \textit{Entities Differentiated by a } Kr}{\#\ \textit{Entities Differentiated by a } K}$$



Evaluation Results

How well this framework can perform on the linkage ?

ex^{R}_{max}		0%	0.5%	1%	2%	3%	4%	5%	10%
Start DB	**DB**	0.39%	100%	100%	100%	100%	100%	100%	100%
	WK	18.67%	601.74%	600.60%	600.77%	600.50%	600.50%	600.50%	599.75%
Start WK	**WK**	6.46%	18.27%	18.24%	18.24%	20.98%	21.28%	21.28%	21.29%
	DB	0.12%	100%	100%	100%	100%	100%	100%	100%

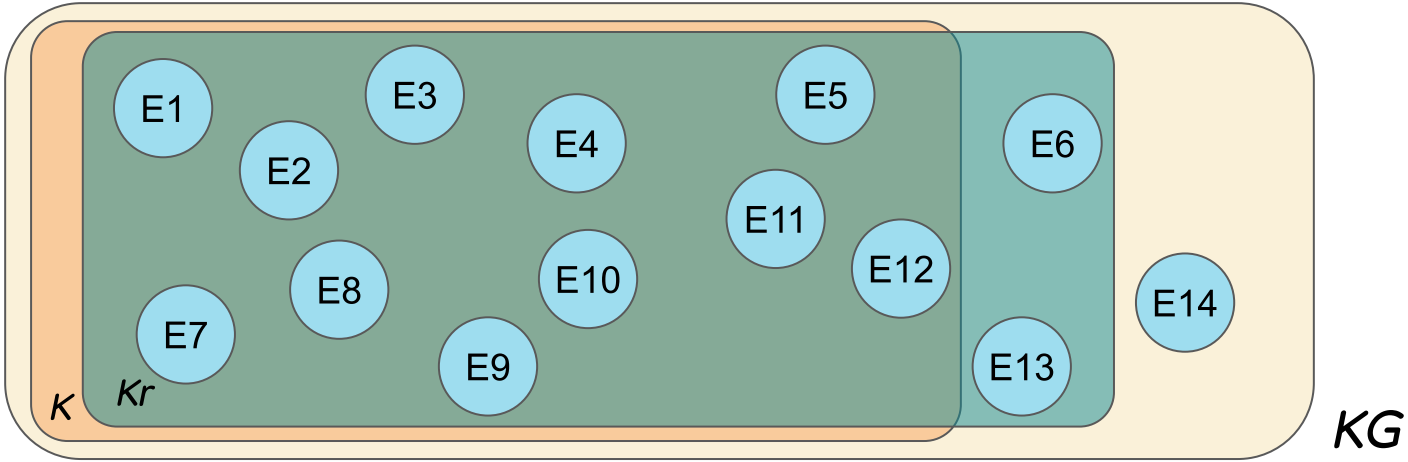
Percentage of entities (among those that are distinguished by a key from the graph) that are distinguishable from every other by at least a key rewritten

Through this metric we observe that **Wikidata** is better than we **anticipated** and this **framework** allows **better results** for **Wikidata** than the **traditional framework.**

Evaluation Results

How well this framework can perform on the linkage ?

$$\frac{\text{\# Entities Differentiated by a } Kr}{\text{\# Entities Differentiated by a } K}$$



Conclusion

- A **faster framework** to perform **key based entity linking.**
- A new definition of key validity based on **relative exception rate**
- A mixed result for the linking problem.
    - Close to the number of entities that can be differentiated through keys
    - Not enough entities to fully link all the entities

Future works

- **Deep study** of the behavior of **keys** under this new **relative exception rate**

- Use of a Catalog (**KeyMap**) to have an even **faster framework:**
  - We could store the keys found on KG1 and directly apply them to KGx, given by a new user

- Define an **unsupervised framework** for **machine learning based methods :**
  - We could overcome the **seed issue** by finding them through **our framework.**

Étude de transférabilité des clés pour le liage de données entre graphes de connaissances

Thibaut Soulard, Fatiha Saïs, Joe Raad, Gianluca Quercini
LISN, CNRS (UMR 9015), Université Paris Saclay, France

LABORATOIRE INTERDISCIPLINAIRE DES SCIENCES DU NUMÉRIQUE

Ingénierie des Connaissances
04 juillet 2023, Strasbourg, France

	Step	0	0.5%	1%	2%	3%	4%	5%	10%
Common DB & WK	*Key discovery DB*	313	301	306	296	301	301	303	307
	Key discovery WK	8.6	6.6	6.7	6.7	6.7	6.6	6.6	6.7
	Total	**322**	**308**	**313**	**302**	**308**	**308**	**310**	**314**
New Start DB	*Key discovery DB*	313	301	306	296	301	301	303	307
	Verification	11.7	11.7	11.7	11.7	11.9	11.9	11.9	11.7
	Total	**324.9**	**313.1**	**318.0**	**308.0**	**313.4**	**313.5**	**315.4**	**319.1**
New Start WK	*Key discovery WK*	8.6	6.6	6.7	6.7	6.7	6.6	6.6	6.7
	Verification	5.9	5.8	6.0	6.0	6.0	6.0	6.0	6.0
	Total	**14.6**	**12.5**	**12.7**	**12.7**	**12.6**	**12.7**	**12.7**	**12.6**

*Running time in **minute** per scenario and step*

Results

Is the validity check necessary ?

Reduce the necessity with a better Property Alignment. Especially because a transformer approach has a worst precision on abbreviation/unknown words.

"BAnQ author ID" ⇔ *"**Bibliothèque et Archives nationales du Québec** author ID"*

	Precision
Transformer (MPNET)	86.3%

But we could always be have difference in the data and thus we may still want this step.