Key discovery in the Semantic Web

Danai Symeonidou

Researcher (CR2)

INRA Montpellier

November, 24th 2015

Web of Data

- ***Semantic Web**: "An extension of the Web that provides a common framework for sharing and reusing data."* W3C

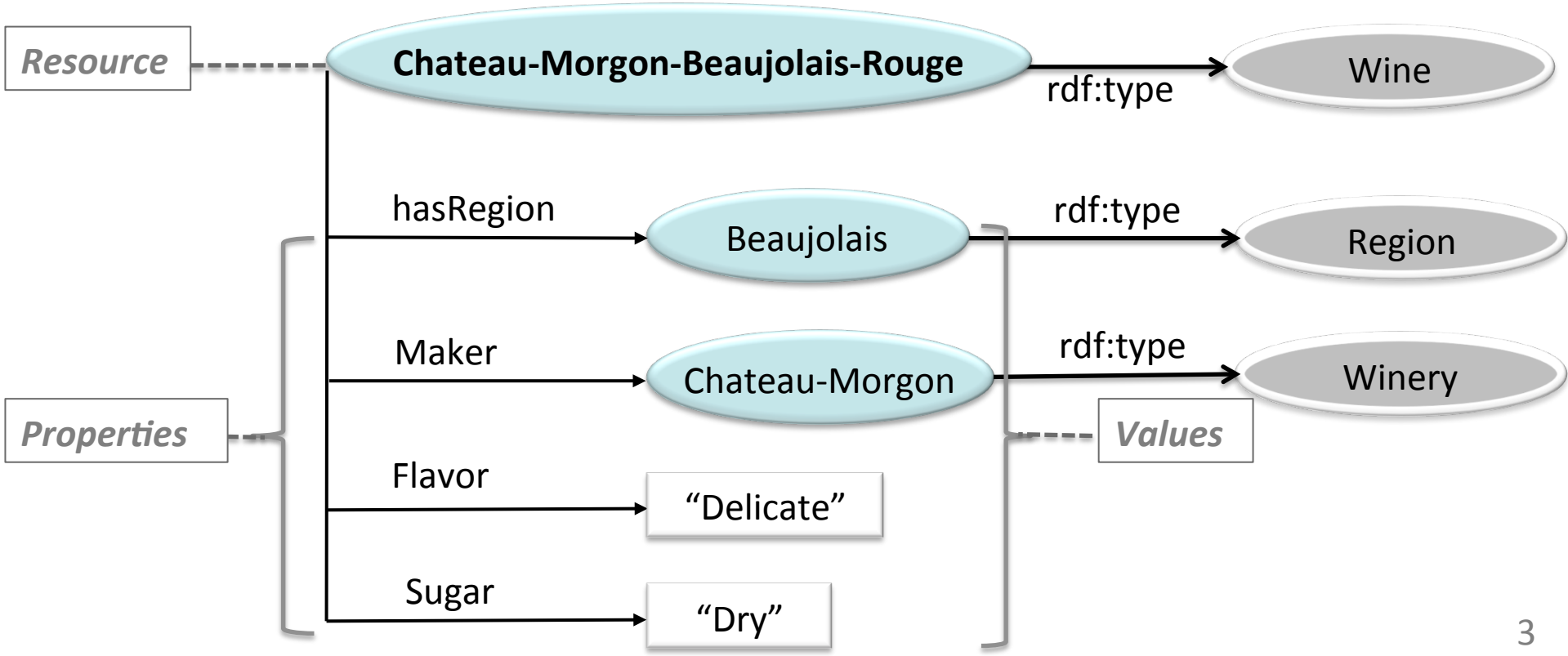
- ***Web of Data:** "Data can be processed by machines."* W3C

- ***Semantic Web technologies:** RDF, OWL, SPARQL*
  - *Uniform format and structured data*

Web of Data: RDF

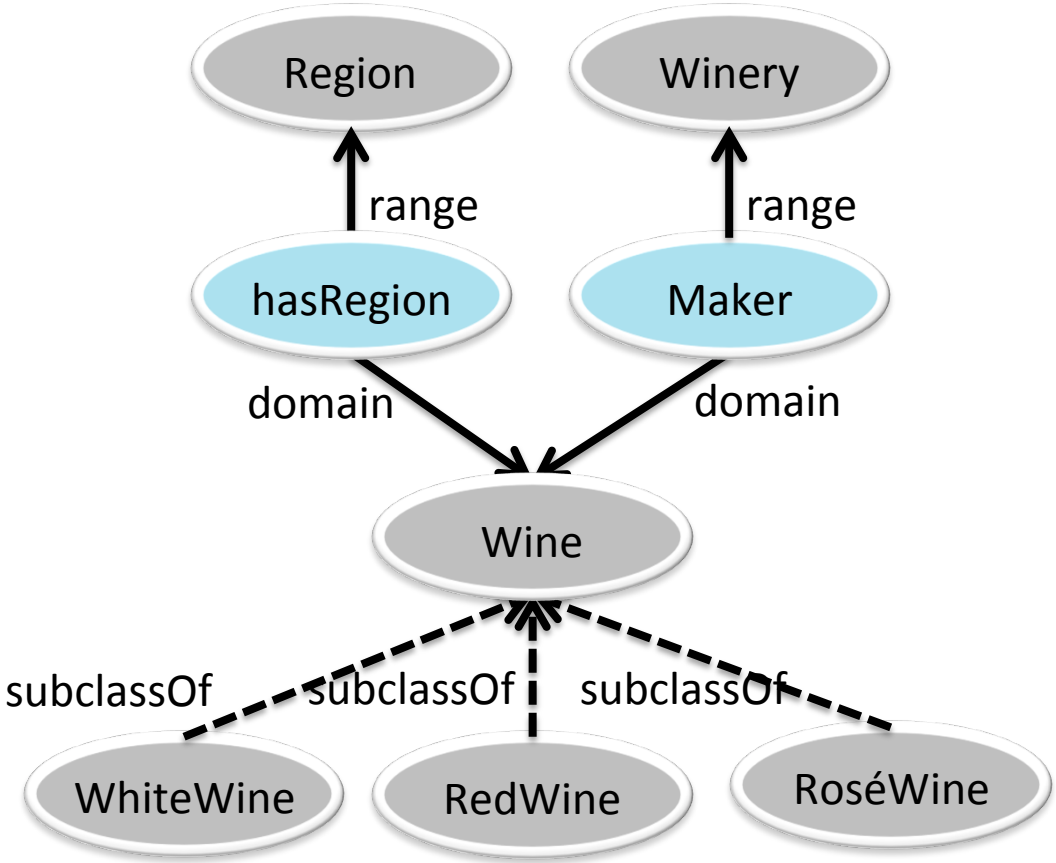
- RDF fact: **property (resource, value)**

  Ex. rdf:type(Chateau-Morgon-Beaujolais-Rouge, Wine)
  hasRegion(Chateau-Morgon-Beaujolais-Rouge, Beaujolais)



Web of Data: Ontology

- Ontologies provide a vocabulary used to represent RDF data

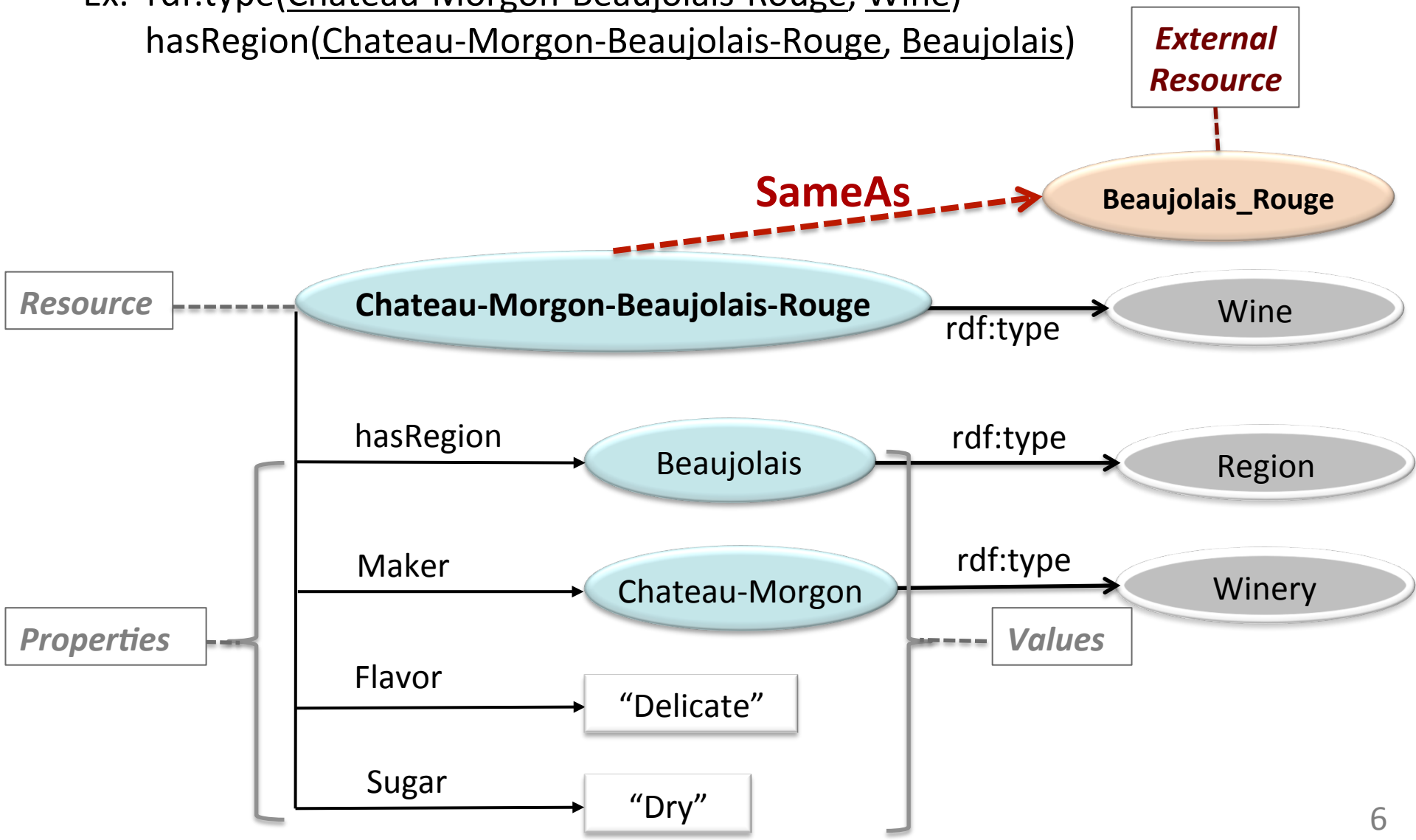


Web of Data

Is the use of RDF and ontologies enough to obtain a Web of Data?

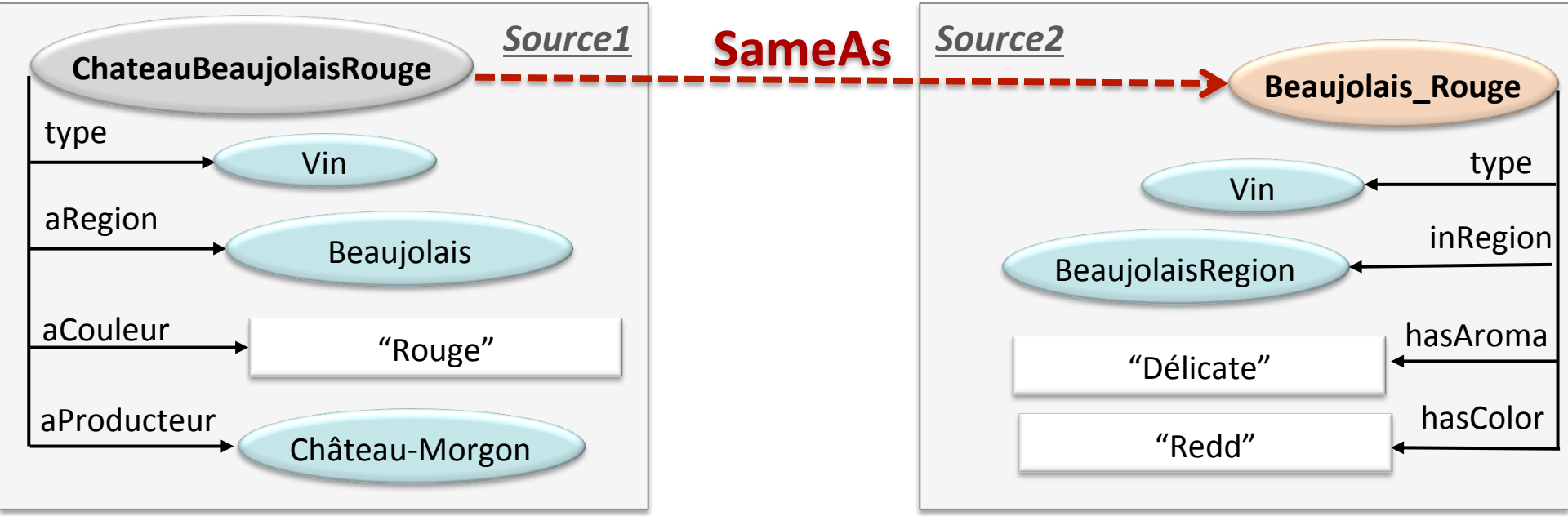
Web of Data: SameAs links

Ex. rdf:type(Chateau-Morgon-Beaujolais-Rouge, Wine)
hasRegion(Chateau-Morgon-Beaujolais-Rouge, Beaujolais)



Data Linking: SameAs links

- **SameAs links**: connect instances of a class referring to the same real world object



More and more data available

- **Hard to define manually sameAs links**

Data Linking approaches

Different criteria can be used to distinguish data linking approaches [FNS11]

- **Instance-based** approaches: exploit property values to link 2 instances / **Graph-based** approaches: propagate similarities, decisions
- **Supervised** approaches : exploit labeled training data given by an expert / **Unsupervised** approaches
- **Knowledge based** approaches : exploit ontology axioms (eg. functional properties, disjunctions) or expert rules
- **Logical** or **Numerical** approaches

Data Linking approaches

Different criteria can be used to distinguish data linking approaches [FNS11]

- **Instance-based** approaches: exploit property values to link 2 instances / **Graph-based** approaches: propagate similarities, decisions
- **Supervised** approaches : exploit labeled training data given by an expert / **Unsupervised** approaches
- **Knowledge based** approaches : exploit ontology axioms (eg. functional properties, disjunctions) or expert rules
- **Logical** or **Numerical** approaches

Most of these approaches use rules to link data

Data Linking using rules

- Linkage Rules
  - Logical Linkage Rules
    - SSN(p1, y) ∧ SSN(p2, y) ➔ sameAs(p1, p2)
  - Complex Linkage Rules
    - max(jaccard(Name(p1, n); Name(p2, m);jarowinkler(address(p1, x); address(p2, y))) > 0.8 ➔ sameAs(p1, p2)

Data Linking using rules

- Linkage Rules
  - Logical Linkage Rules
    - **SSN**(p1, y) ∧ **SSN**(p2, y) ➔ sameAs(p1, p2)

    {SSN}: discriminative property

  - Complex Linkage Rules
    - max(jaccard(**Name**(p1, n); **Name**(p2, m);jarowinkler(**address**(p1, x); **address**(p2, y))) > 0.8 ➔ sameAs(p1, p2)

    {Name, Address}: discriminative property set

Rules contain discriminative properties => keys

OWL2 Key

- OWL (Web Ontology Language)

- **OWL2 Key for a class**: a combination of properties that uniquely identify each instance of a class

$$\forall X, \forall Y, \forall Z_1, \ldots, Z_n, \forall T_1, \ldots, T_m \wedge ce(X) \wedge ce(Y) \bigwedge_{i=1}^{n} (ope_i(X, Z_i) \wedge ope_i(Y, Z_i))$$

$$\bigwedge_{i=1}^{m} (dpe_i(X, T_i) \wedge dpe_i(Y, T_i)) \Rightarrow X = Y$$

hasKey(Person(SSN)) means:

Type(P_1, Person) $\wedge$ type(P_2, Person) $\wedge$ SSN(P_1, y) $\wedge$ SSN(P_2, y) ➔ sameAs(P_1, P_2)

Keys declared by experts for data linking

- Not an easy task:
  - Experts are not aware of all the keys

    Ex. {SSN}, {ISBN} easy to declare

    Ex. {Region, Flavor, Produced} **is it a key for the class wine?**

  - Erroneous keys can be given by experts

  - As many keys as possible
    - More keys => More linking rules

Keys declared by experts for data linking

- Not an easy task:
  - Experts are not aware of all the keys

    Ex. {SSN}, {ISBN} easy to declare

    Ex. {Region, Flavor, Produced} **is it a key for the class wine?**

  - Erroneous keys can be given by experts

  - As many keys as possible
    - More keys => More linking rules

- **Goal: Automatic discovery of keys from the data**

Key Discovery - Related Work

- Key discovery previously studied in **Relational databases**
  - No strategies to treat incomplete data
  - No multivaluation of properties
  - No ontology to take into account
  - No strategies to be scalable in data found on the Web

Semantic Web					
Approach	**Composite keys**	**Complete set of keys**	**OWL2 keys**	**Approximate keys**	**Incomplete data heuristics**
[SAS11]			✓	✓	
[SH11]	✓		✓	✓	
[ADS12]	✓	✓		✓	✓

Key Discovery - Related Work

- Key discovery previously studied in **Relational databases**
  - No strategies to treat incomplete data
  - No multivaluation of properties
  - No ontology to take into account
  - No strategies to be scalable in data found on the Web

Semantic Web					
Approach	**Composite keys**	**Complete set of keys**	**OWL2 keys**	**Approximate keys**	**Incomplete data heuristics**
[SAS11]			✓	✓	
[SH11]	✓		✓	✓	
[ADS12]	✓	✓		✓	✓

- We are the first to propose an approach that fulfills all these characteristics

Problem statement

- How to discover keys in RDF data when
  - They are incomplete?
  - They contain errors?
  - They contain duplicates?
  - They are numerous and described by many properties?

Contributions

- **KD2R**[*]: Key discovery for data linking
  - Complete set of composite keys
  - Keys following the definition of OWL2
  - Incomplete data
  - Ontology semantics (subsumptions)

- **SAKey**[**]: Scalable Almost Key discovery for data linking
  - Complete set of composite keys
  - Keys following the definition of OWL2
  - Incomplete data
  - Ontology semantics (subsumptions)
  - **Erroneous data**
  - **Duplicates**
  - **Large datasets**

** Journal of Web Semantics (JWS), 2013*
*** International Semantic Web Conference (ISWC), 2014*

Key discovery in incomplete data

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2,i3
i2	Tompson	Maria	
i3	David	George	i2, i4
i4	Solgar	Michel	

- hasFriend(i1,i4) …. ?
- hasFriend(i2, i3) …. ?
- firstName(i1, Elodie) … ?

…

Key discovery in incomplete data

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2,i3
i2	Tompson	Maria	
i3	David	George	i2, i4
i4	Solgar	Michel	

- **Optimistic heuristic**
  - All Properties ➔ only given values are considered
- **Pessimistic heuristic**
  - Not instantiated property ➔ value possibly one of the existing ones
  - Instantiated property ➔ only given values are considered

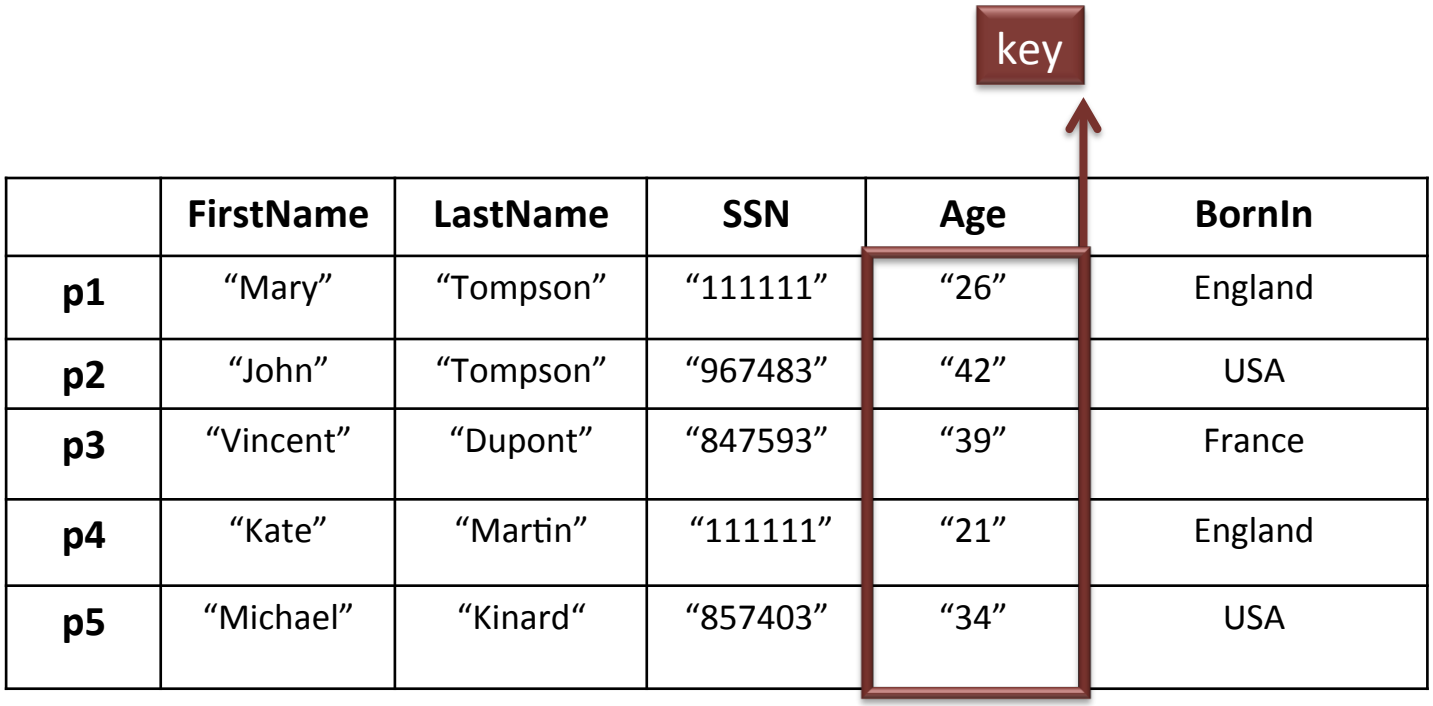
Key discovery in incomplete data

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2,i3
i2	Tompson	Maria	**i2, i3, i4**
i3	David	George	i2, i4
i4	Solgar	Michel	**i2, i3, i4**

- **Optimistic heuristic**
  - All Properties ➔ only given values are considered
- **Pessimistic heuristic**
  - Not instantiated property ➔ value possibly one of the existing ones
  - Instantiated property ➔ only given values are considered

Key discovery in erroneous data

- How can we discover keys in the presence of errors and/or duplicates?



	FirstName	LastName	SSN	Age	BornIn
p1	“Mary”	“Tompson”	“111111”	“26”	England
p2	“John”	“Tompson”	“967483”	“42”	USA
p3	“Vincent”	“Dupont”	“847593”	“39”	France
p4	“Kate”	“Martin”	“111111”	“21”	England
p5	“Michael”	“Kinard“	“857403”	“34”	USA

Key discovery in erroneous data

- How can we discover keys in the presence of errors and/or duplicates?
- When RDF data contain errors and/or duplicates keys can be lost

	FirstName	LastName	SSN	Age	BornIn
p1	"Mary"	"Tompson"	**"111111"**	"26"	England
p2	"John"	"Tompson"	"967483"	"42"	USA
p3	"Vincent"	"Dupont"	"847593"	"39"	France
p4	"Kate"	"Martin"	**"111111"**	"21"	England
p5	"Michael"	"Kinard"	"857403"	"34"	USA

Key discovery in erroneous data

- Discovery of sets of properties that are not keys due to few exceptions
- **Exception of a key *P*:** an instance that shares values with another instance for a given set of properties *P*
  - p1 and p4 are exceptions for {SSN}

	FirstName	LastName	SSN	Age	BornIn
p1	"Mary"	"Tompson"	**"111111"**	"26"	England
p2	"John"	"Tompson"	"967483"	"42"	USA
p3	"Vincent"	"Dupont"	"847593"	"39"	France
p4	"Kate"	"Martin"	**"111111"**	"21"	England
p5	"Michael"	"Kinard"	"857403"	"34"	USA

n-almost keys

- **Exception Set E_P:** set of exceptions for *P*
  - E_{SSN} = {p1, p4}
- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {SSN} is a 2-almost key

	FirstName	LastName	SSN	Age	BornIn
p1	"Mary"	"Tompson"	**"111111"**	"26"	England
p2	"John"	"Tompson"	"967483"	"42"	USA
p3	"Vincent"	"Dupont"	"847593"	"39"	France
p4	"Kate"	"Martin"	**"111111"**	"21"	England
p5	"Michael"	"Kinard"	"857403"	"34"	USA

- *n* value is declared by an expert

Almost key discovery strategy

- The key discovery is a #P-Hard problem
  - Optimization techniques are needed to scale

- **Naive automatic way to discover almost keys**
  - Examine all the possible combinations of properties
  - Scan all instances for each candidate almost key

  Example: Class described by 15 properties → 2^{15} = 32768 candidate almost keys

- Discover almost keys efficiently by:
  - Reducing the combinations
  - Partially scanning the data

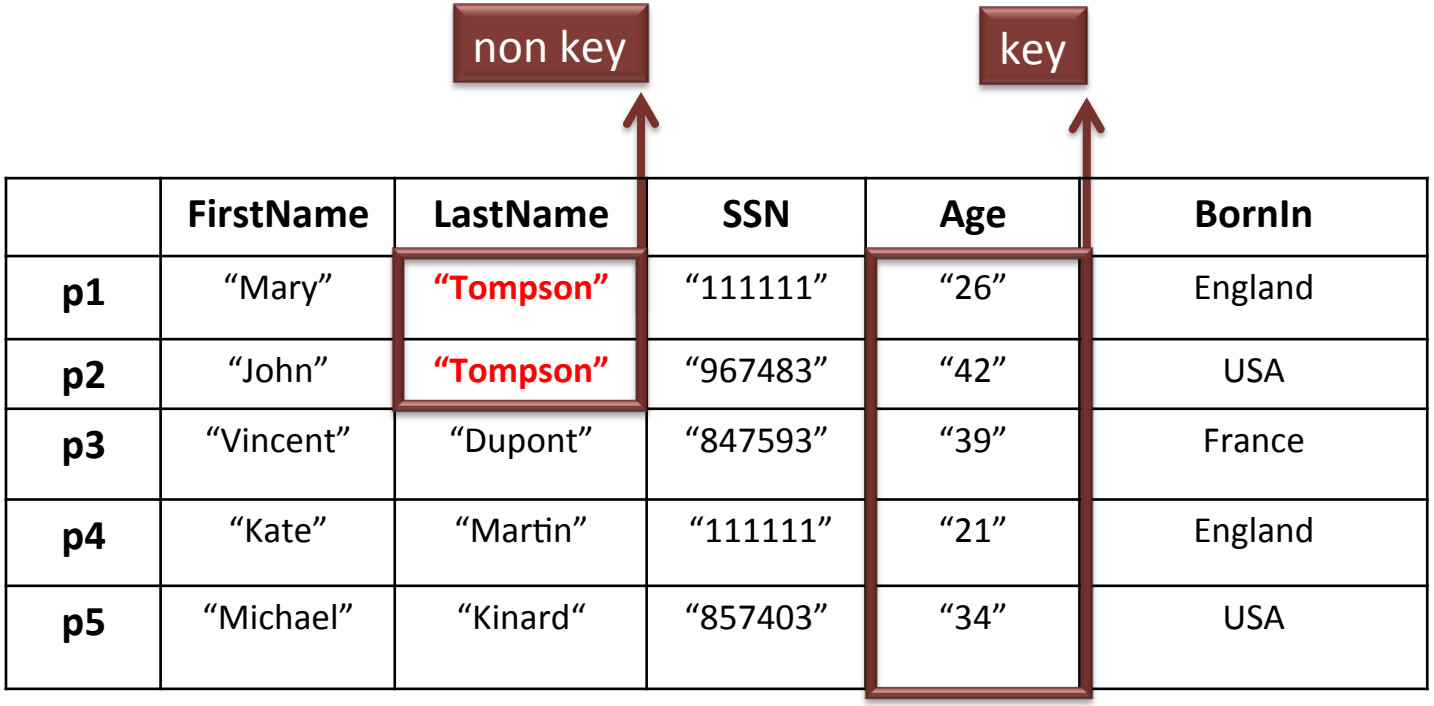
Almost key discovery strategy

- Discover sets of properties that are not keys, i.e., non keys first
- Why discovering non keys first allow to partially scan the data?

	FirstName	LastName	SSN	Age	BornIn
p1	“Mary”	“Tompson”	“111111”	“26”	England
p2	“John”	“Tompson”	“967483”	“42”	USA
p3	“Vincent”	“Dupont”	“847593”	“39”	France
p4	“Kate”	“Martin”	“111111”	“21”	England
p5	“Michael”	“Kinard“	“857403”	“34”	USA

Almost key discovery strategy

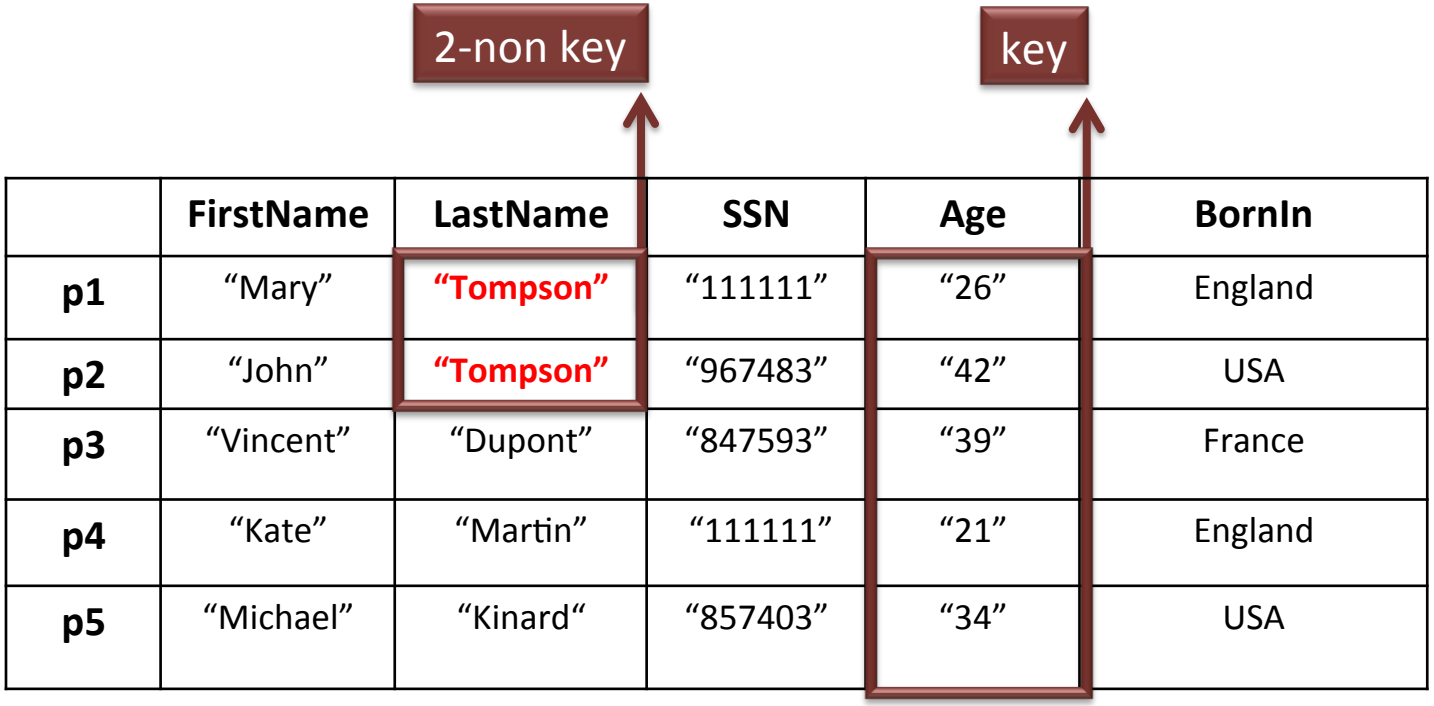
- Discover sets of properties that are not keys, i.e., non keys first
- Why discovering non keys first allow to partially scan the data?



	FirstName	LastName	SSN	Age	BornIn
p1	"Mary"	**"Tompson"**	"111111"	"26"	England
p2	"John"	**"Tompson"**	"967483"	"42"	USA
p3	"Vincent"	"Dupont"	"847593"	"39"	France
p4	"Kate"	"Martin"	"111111"	"21"	England
p5	"Michael"	"Kinard"	"857403"	"34"	USA

Almost key discovery strategy

- Discover sets of properties that are not keys, i.e., non keys first
- Why discovering non keys first allow to partially scan the data?



	FirstName	LastName	SSN	Age	BornIn
p1	"Mary"	**"Tompson"**	"111111"	"26"	England
p2	"John"	**"Tompson"**	"967483"	"42"	USA
p3	"Vincent"	"Dupont"	"847593"	"39"	France
p4	"Kate"	"Martin"	"111111"	"21"	England
p5	"Michael"	"Kinard"	"857403"	"34"	USA

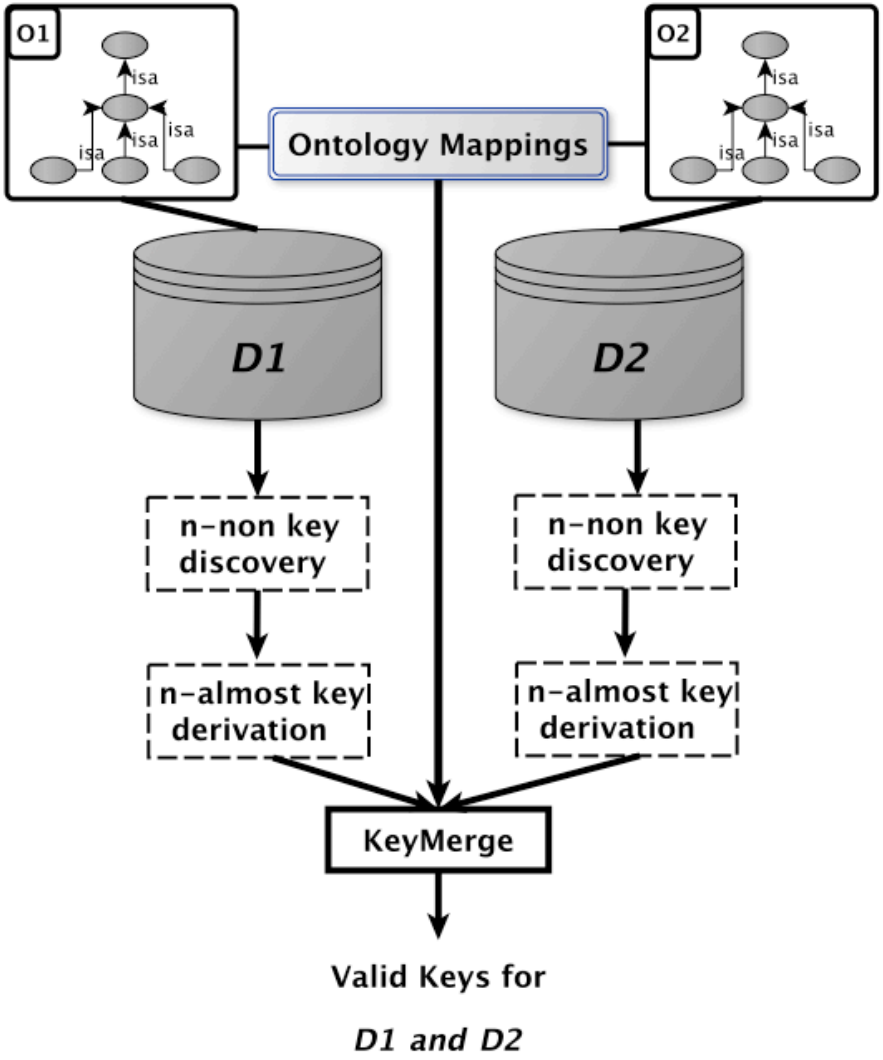
- ***n*-non keys:** set of properties where $|E_p| \geq n$

Scalability of SAKey

- **Scalability in *n*-non key discovery**
  - **Inclusion pruning**
    - Discovery of dependencies between data
  - **Seen intersection pruning**
    - Avoiding already explored sets of instances
  - **Irrelevant intersection pruning**
    - Ordering of instances to avoid useless computations
  - **Antimonotonic pruning**
    - All the subsets of a *n*-non key are at least *n*-non keys

- **Scalability in *n*-almost key derivation**
  - **Efficient derivation of minimal n-almost keys from maximal (n+1)-non keys**

Key discovery in several datasets



D1: {firstName, LastName}
D2: {DateOfBirth}

D12: {firstName, LastName, DateOfBirth}

Experiments

- Evaluation of the quality of discovered keys
  - Evaluation of discovered keys by experts
  - Keys in Data Linking
- Scalability of SAKey
- Selected datasets
  - DBpedia, YAGO, INA, ABES, ChefMoz, GFT - Real data
  - OAEI 2010, OAEI 2011, OAEI 2013 - Synthetic data

Evaluation of keys by experts

- Discovered keys were shown to experts

- Datasets
  - INA (National Audiovisual Institute)
  - ABES (Bibliographic Agency for Higher Education)

- Conclusion
  - Experts were not always able to decide whether a discovered key was referring to a real key

Keys in Data Linking

- Data linking using
  - Discovered keys
  - Expert keys
  - No keys

- Evaluation of linking using
  - **Recall**: ratio of relevant retrieved links to the total number of relevant links
  - **Precision**: ratio of relevant retrieved links to the total number of retrieved links
  - **F-Measure**: harmonic mean of precision and recall

- Datasets: OAEI 2010, OAEI 2011, OAEI 2013, ChefMoz, GFT

- Conclusion
  - Linking results using discovered keys are better than expert keys and no keys
  - Exceptions provide more correct links without significantly decreasing the precision

Example: Data Linking using almost keys

- OAEI 2013 - Person
  - BirthName, BirthDate, award, comment, label, BirthPlace, almaMater, doctoralAdvisor

	Almost keys	Recall	Precision	F-Measure
0-almost key	{BirthDate, award}	9.3%	100%	17%
2-almost key	{BirthDate}	32.5%	98.6%	49%

# exceptions	Recall	Precision	F-measure
0, 1	25.6%	100%	41%
2, 3	47.6%	98.1%	64.2%
4, 5	47.9%	96.3%	63.9%
6, ..., 16	48.1%	96.3%	64.1%
17	49.3%	82.8%	61.8%

Scalability of SAKey

- Evaluate the scalability of SAKey on 9 datasets (DBpedia, Yago, OAEI, etc.)

- Conclusion
  - SAKey can up to million triples thanks to pruning and filtering strategies
    - DB:Person, **biggest class of DBpedia** with
      - 8 million triples,
      - 9 hundred thousand instances,
      - 508 properties

Conclusion

- **Key discovery taking into account:**
  - Incomplete data
    - Two heuristics to deal with incomplete data: optimistic/pessimistic keys
  - Erroneous data, duplicates
    - *n*-almost keys: keys with at most *n* exceptions
  - Being scalable thanks to:
    - Filtering and pruning strategies
    - Scalable key derivation approach
  - Experiments show the scalability of SAKey and the relevance of almost keys in data linking

Conclusion

- **Key discovery taking into account:**
  - Incomplete data
    - Two heuristics to deal with incomplete data: optimistic/pessimistic keys
  - Erroneous data, duplicates
    - *n*-almost keys: keys with at most *n* exceptions
  - Being scalable thanks to:
    - Filtering and pruning strategies
    - Scalable key derivation approach
  - Experiments show the scalability of SAKey and the relevance of almost keys in data linking

Thank you for your attention!

Publications

- **International Journals**
  - Nathalie Pernelle, Fatiha Saïs, Danai Symeonidou. *An automatic key discovery approach for data linking*. **Journal of Web Semantics**, Volume 23 pages 16–30, 2013.

- **International Conferences/Workshops/Demos**
  - Luis Galárraga, Danai Symeonidou, Jean-Claude Moissinac, *Rule Mining for Semantifying Wikilinks,* Linked Data On the Web workshop **(LDOW, WWW 2015)**
  - Ziad Ismail, Danai Symeonidou, Fabian Suchanek, *DIVINA: Discovering vulnerabilities of Internet accounts,* Demo Paper, World Wide Web **(WWW 2015)**
  - Danai Symeonidou, Vincent Armant, Nathalie Pernelle, Fatiha Saïs. *SAKey: Scalable Almost Key discovery in RDF data*. 13th International Semantic Web Conference **(ISWC 2014)**. To appear in ISWC 19-23 October 2014, Trento, Italy.
  - Manuel Atencia, Michel Chein, Madalina Croitoru, Michel Leclere Jerome David, Nathalie Pernelle, Fatiha Saïs, Francois Scharffe, Danai Symeonidou. *Defining key semantics for the rdf datasets: Experiments and evaluations.* International Conferences on Conceptual Structures (**ICCS 2014)**, Iasi, Romania.
  - Symeonidou, D., Pernelle, N. and Saïs, F. (2013). *Discovering Keys in RDF/OWL Dataset with KD2R.* 2nd International workshop on Open Data (**WOD 2013**), Demo paper, Paris, France
  - Symeonidou, D., Pernelle, N. and Saïs, F. (2011). *KD2R: a Key Discovery method for semantic Reference Reconciliation in OWL2*, Workshop on Semantic Web & Web Semantics (**SWWS 2011**), 392–401, Heraklion, Greece

- **National Conferences**
  - Nathalie Pernelle, Danai Symeonidou, Fatiha Sais, *C-SAKey : une approche de découverte de clés conditionnelles dans des données RDF*, 26es Journées francophones d'ingénerie des Connaissances **(IC 2015)**
  - Chein, M., Croitoru, M., Leclère, M., Pernelle, N., Saïs, F. and Symeonidou, D. (2014). *Defining Key Semantics for the Semantic Web (A Theoretical View),* 25es Journées francophones d'ingénerie des Connaissances (**IC 2014**) Clemont Ferrard, France
  - Danai Symeonidou, Vincent Armant, Nathalie Pernelle, Fatiha Saïs. *SAKey: Scalable Almost Key discovery in RDF data.* Bases de Données Avancées (**BDA 2014**), 14-17 Octobre 2014, Grenoble, France.

References

- **[SBHR06]** Yannis Sismanis, Paul Brown, Peter J. Haas, and Berthold Reinwald. Gordian: efficient and scalable discovery of composite keys. In *Proceedings of the 32nd International conference Very Large Data Bases (VLDB),* VLDB '06, pages 691–702. VLDB Endowment, 2006.
- **[SAS11]** Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *The Proceedings of the VLDB Endowment(PVLDB),* 5(3):157–168, 2011.
- **[FNS11]** Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking for the semantic web. Int. J. Semantic Web Inf. Syst., 7(3):46–76, 2011.
- **[SH11]** Dezhao Song and Jeff Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In *Proceedings of the 10th International Semantic Web Conference(ISWC) - Volume Part I,* ISWC'11, pages 649–664, Berlin, Heidelberg, 2011. Springer-Verlag.
- **[AN11]** Ziawasch Abedjan and Felix Naumann. Advancing the discovery of unique column combinations. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, pages 1565– 1570, New York, NY, USA, 2011. ACM.
- **[VLM12]** S. Link V. Le and M. Memari. Schema- and data-driven discovery of sql keys. *JCSE,* 6(3):193–206, 2012.
- **[ADS12]** Manuel Atencia, Jérôme David, and François Scharffe. Keys and pseudo- keys detection for web datasets cleansing and interlinking. In *EKAW,* pages 144–153, 2012.
- **[KLL13]** Henning Köhler, Uwe Leck, and Sebastian Link. Possible and certain sql keys. Technical report, Centre for Discrete Mathematics and Theoretical Computer Science, 2013.
- **[HJAQR+13]** A. Heise, Jorge-Arnulfo, Quiane-Ruiz, Z. Abedjan, A. Jentzsch, and F. Nau- mann. Scalable discovery of unique column combinations. *VLDB,* 7(4):301– 312, 2013.