BLOOMS

Ontology Alignment for Linked Open Data

Outline

- Introduction
- Problem definition
- BLOOMS approach
- Evaluation
- Future work

Introduction

- Linked data
- Ontology alignment

Linked data

- Increasing need for structured data
  - Amazon ecosystem of affiliates
  - Google and Yahoo! shoping engines
  - TheyWorkForYou
- HTML is oriented towards structuring text documents
  - Data is mixed with text
  - Hard for machines to extract structured data
  - Microformats too restricted!

Linked data

- Internet is therefore the web of documents
  - Documents linked with <*a href*>
  - Search engines use crawlers to create web page index
  - Web publishers register a page with each SE
- Goal is to create the web of data
  - RDF describes concepts and relations between concepts
  - Concepts from different APIs are linked explicitly
  - "myBook *forSaleIn* thatBookshop *locatedIn* myCity"

Ontology alignment

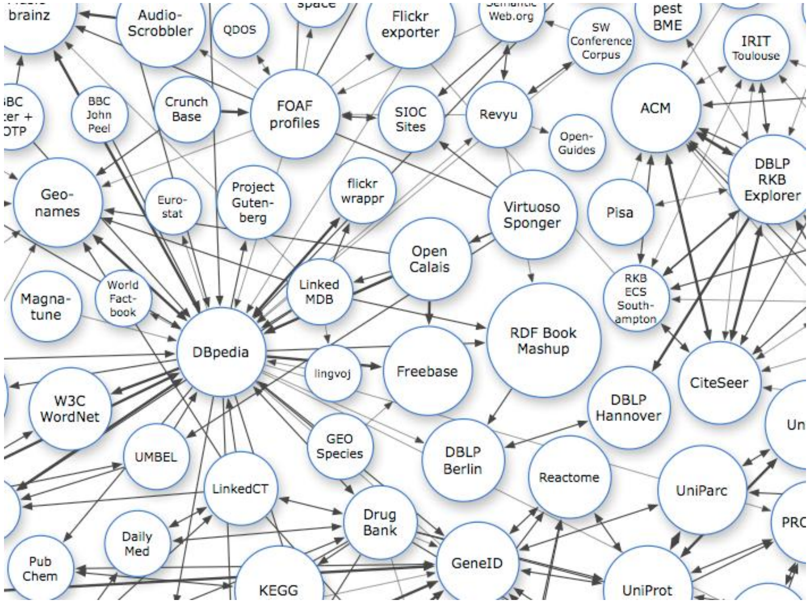
- Proc. of finding correspondences between concepts
- Today concepts are very diverse
  - Every system has its own vocabulary
  - Ontologies are developed independently
- Need to integrate heterogenous dbs
- Tools find classes that are semantically equivalent
  - Eg. “Truck” and “Lorry”
- These tools are called ontology alignment tools

Problem definition

- State of the web
- Central issues

State of the web

- LOD community effort resulted in "The web of data"
  - Contains several billion RDF triples
  - Very diverse



Part of the LOD cloud, July, 2009

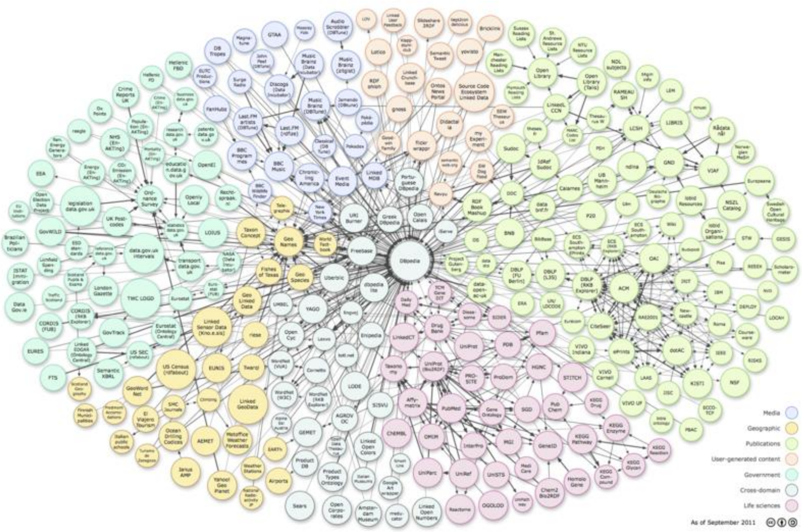
Central issues

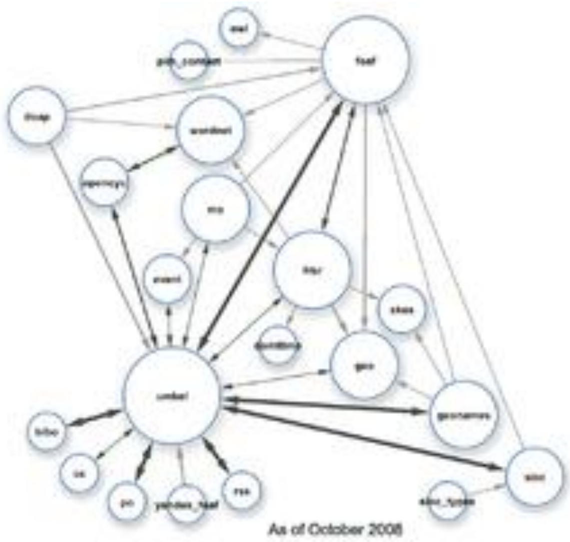
- Interlinks between datasets still relatively scarce
  - Mainly on the instance level
  - Using owl:sameAs
- Schema-level taxonomy info even more scarce
  - rdfs:subClassOf
  - In particular, lack of links between different schemas
- Example:
  - An artist on DBpedia
  - Composer on LinkedMDB

Central issues

Instance-level linkages

Class linkages





BLOOMS approach

1. Pre-processing of input ontologies
2. Construction of BLOOMS forest
3. Comparison of BLOOMS forests
4. Post-processing

BLOOMS approach

- State-of-art alignment systems fail on LOD datasets
- BLOOMS uses bootstrapping approach
  - Wikipedia category hierarchy
  - Already on the LOD cloud
  - Noisy community-generated data
- Goal is to create taxonomy links between A and B
  - A rdfs:subClassOf B
  - B rdfs:subClassOf A
  - A owl:equivalentClass B
  - none of the above

BLOOMS approach

- Centered around constructing a forest for class C
  - For class C, *Tc* is "*BLOOMS forest for C*"
  - Represents a selection of Wikipedia supercategories
  - Comparison of forests T_C and T_B yields results
- Running example are class names
  1. Event (DBpedia dataset)
  2. JazzFestival (Music Ontology dataset)

Pre-processing

- Normalization of Class names C
  - Replacing underscores and hyphens by spaces
  - Splitting by capital letters
  - Stop word removal
- The result is a normalized string C'
- In our running example
  1. C = JazzFestival, C' = Jazz Festival
  2. D = Event, D' = Event

Construction of the BLOOMS forest

- We invoke Wikipedia Web Service for C'
  - The results is the Wc Wikipedia set of pages
  - If only one page is returned then Tc is a tree
  - If we get disambig. page then all pages are added
- The result set Wc is called *senses* for C
- For each sense *s* ∈ Wc we create Ts ∈ Tc:
  - Root is s
  - Children of s are all categories for that page
  - Children of category C are super-categories of C
  - Tree is cut at level 4

Construction of the BLOOMS forest

$T_{\text{jazz Festival}}$

T_{Event}

Comparison of BLOOMS forests

- We do comparison of concept names C and D
- We compare each Ts ∈ T_C and Tt ∈ T_D
- Function o(Ts, Tt) is a real number overlap measure
  - Remove from Ts nodes that have parent in Tt
  - Removed nodes do not reveal any new info
  - Calculate overlap info with the formula:

$$o(Ts, Tt) = \frac{n}{k-1}$$

  - n is number of nodes in Ts' that appear in Tt and k is the total number of nodes in Ts'

Comparison of BLOOMS forests

- Alignment is calculated as follows:
  - C owl:equivalentClass D if: $T_s = T_t \mid T_s \in T_C, T_t \in T_D$
  - For some pre-defined threshold x if:

$$\min\{o(T_s, T_t), o(T_t, T_s)\} \geq x$$

    - C rdfs:subClassOf D if: $o(T_s, T_t) \geq o(T_t, T_s)$
    - D rdfs:subClassOf C if: $o(T_s, T_t) \leq o(T_t, T_s)$
- For our running example we have
  - $o(T_{Event}, T_{Jazz\ Festival}) > o(T_{Jazz\ Festival}, T_{Event})$
  - The result is: Jazz Festival rdfs:subClassOf Event

Post-processing

- Invoke Alignment API
  - Find alignments between original input ontologies
  - Keep only the ones with confidence value at least 0.95
  - Add them to the results previously obtained
- Invoke a reasoner
  - Find inferred alignments
  - In our case Jena
- Output alignments in Alignment API format

Evaluation

- General purpose ontology matching
- LOD schema integration
- Related Work

General purpose ontology matching

- Run on OAEI benchmarks
- Compared to other state of the art systems
  - RiMOM
  - AROMA
- BLOOMS input parameters:
  - x = 0.8 for same domain ontologies
  - x = 0.6 where one was an abstract (Dbpedia) ontology
- Two tracks
  - Benchmark: test equivalence
  - Oriented matching: subclass relationships

General purpose ontology matching

Ontology Alignment Initiative—Benchmark Track												
	S-Match		OMViaUO		Alignment API		BLOOMS		AROMA		RiMoM	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Recall	Prec	Rec
1XX	0.11	1	0.26	0.37	0.59	0.96	0.71	1	1	1	1	1
2XX	0.1	0.2	0.21	0.31	0.3	0.54	0.38	0.49	0.88	0.65	0.93	0.81
3XX	0.1	0.2	0.28	0.28	0.45	0.77	0.62	0.84	0.80	0.76	0.81	0.82
Avg.	0.1	0.46	0.25	0.33	0.45	0.76	0.57	0.78	0.88	0.81	0.91	0.88

Ontology Alignment Initiative—Oriented Matching Track												
	A-API		OMV		S-Match		AROMA		RiMoM		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
1XX	0	0	0.02	0.06	0.01	0.71	NaN	0	1	1	1	1
2XX	0	0	0.01	0.03	0.05	0.30	0.84	0.08	0.67	0.85	0.52	0.51
3XX	0.01	0.03	0.02	0.047	0.01	0.14	0.72	0.11	0.59	0.81	1	0.84
Avg.	0.00	0.01	0.02	0.04	0.03	0.38	0.63	0.07	0.75	0.88	0.84	0.78

LOD Schema Alignment

- No established benchmarks
- Human experts created reference alignments
  - Subclass relations
  - Equivalence relations
- Chosen datasets cover significant LOD portion
- Using only publicly available schemas
  - In order to avoid unfair advantage
  - LinkedMDB for instance did not make schema available

LOD datasets

Table 3. LOD datasets=LOD datasets utilizing this schema, D=taxonomic depth, # C=number of classes, Linked datasets=LOD datasets they are linked to at the instance level

Schema	**LOD datasets**	**D**	**# C**	**Linked datasets**
DBpedia[17]	DBpedia	4	204	Geonames, US Census, Freebase
Geonames[18]	Geonames, Geospecies	2	11	DBpedia, Jamendo, FOAF Profiles
Music Ontology[19]	Jamendo, Music Brainz, DBTunes	4	136	GovTrack, DBpedia, Geonames
BBC Program[20]	BBC Programs, BBC Music	4	100	BBC Music, BBC Playcount Data
FOAF Profiles[21]	FOAF, Music Brainz	3	16	Crunch Base, QDOS, SIOC Sites
SIOC[22]	DBpedia, LinkedMDB	2	14	Virtuoso Sponger, FOAF Profiles, SemanticWeb.org
AKT Reference Ontology[23]	ACM, DBLP	5	17	Pisa, IEEE, eprints
Semantic Web Conference Ontology[24]	SW Conference Corpus	5	177	SemanticWeb.org, Revyu

LOD results

Table 4. Results of various systems for LOD Schema Alignment. Legends: Prec=Precision, Rec=Recall, M=Music Ontology, B=BBC Program Ontology, F=FOAF Ontology, D=DBpedia Ontology, G=Geonames Ontology, S=SIOC Ontology, W=Semantic Web Conference Ontology, A=AKT Portal Ontology, err=System Error, NA=Not Available.

Linked Open Data Schema Ontology Alignment												
	Alignment API		OMViaUO		RiMoM		S-Match		AROMA		BLOOMS	
Test	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
M,B	0.4	0	1	0	err	err	0.04	0.28	0	0	0.63	0.78
M,D	0	0	0	0	err	err	0.08	0.30	0.45	0.01	0.39	0.62
F,D	0	0	0	0	err	err	0.11	0.40	0.33	0.04	0.67	0.73
G,D	0	0	0	0	err	err	0.23	1	0	0	0	0
S,F	0	0	0	0	0.3	0.2	0.52	0.11	0.30	0.20	0.55	0.64
W,A	0.12	0.05	0.16	0.03	err	err	0.06	0.4	0.38	0.03	0.42	0.59
W,D	0	0	0	0	err	err	0.15	0.50	0.27	0.01	0.70	0.40
Avg.	0.07	0.01	0.17	0	NA	NA	0.17	0.43	0.25	0.04	0.48	0.54

Related work

- First work using noisy categorization for matching
- Previously, it was used for taxonomy restructuring
- Gen. algorithm for DB schema matching done in [4]
- UMBEL is a notable reference point for LOD schema

Future work

Future work

- Intention to identify other kinds of relationships
  - Partonomical relationships
  - Disjointness
- Release upper level ontology for LOD
  - Based on SUMO or DOLCE
  - Added input of BLOOMS
- Test on other platforms
  - OWL-API
  - Other reasoner then Jena

References

1. Ontology Alignment for Linked Open Data, Jain et.al. Kno.e.sis Center, Wright State University
2. Linked Data: Evolving the Web into a Global Data Space, Tom Heath, Christian Bizer, ISBN: ISBN: 9781608454310 (ebook ISBN)
3. Linked Data - The Story So Far, Bizer et.al.
4. Nikolov, A., Uren, V.S., Motta, E., Roeck, A.N.D.: Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In: G´omez-P´erez, A.,Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 332–346. Springer, Heidelberg (2009)
5. Semantic Web, Vujovic, Neuhold, Fankhauser, Niederee, Milutinovic)

Thank you for your attention!

BLOOMS, The Ontology Alignment for LOD

Azarić Bogdan 11/3035,
bogdan.azaric@gmail.com