Ontology matching and Data Interoperability using community generate data

Prateek Jain
Research Staff Member
IBM TJ Watson Research Center

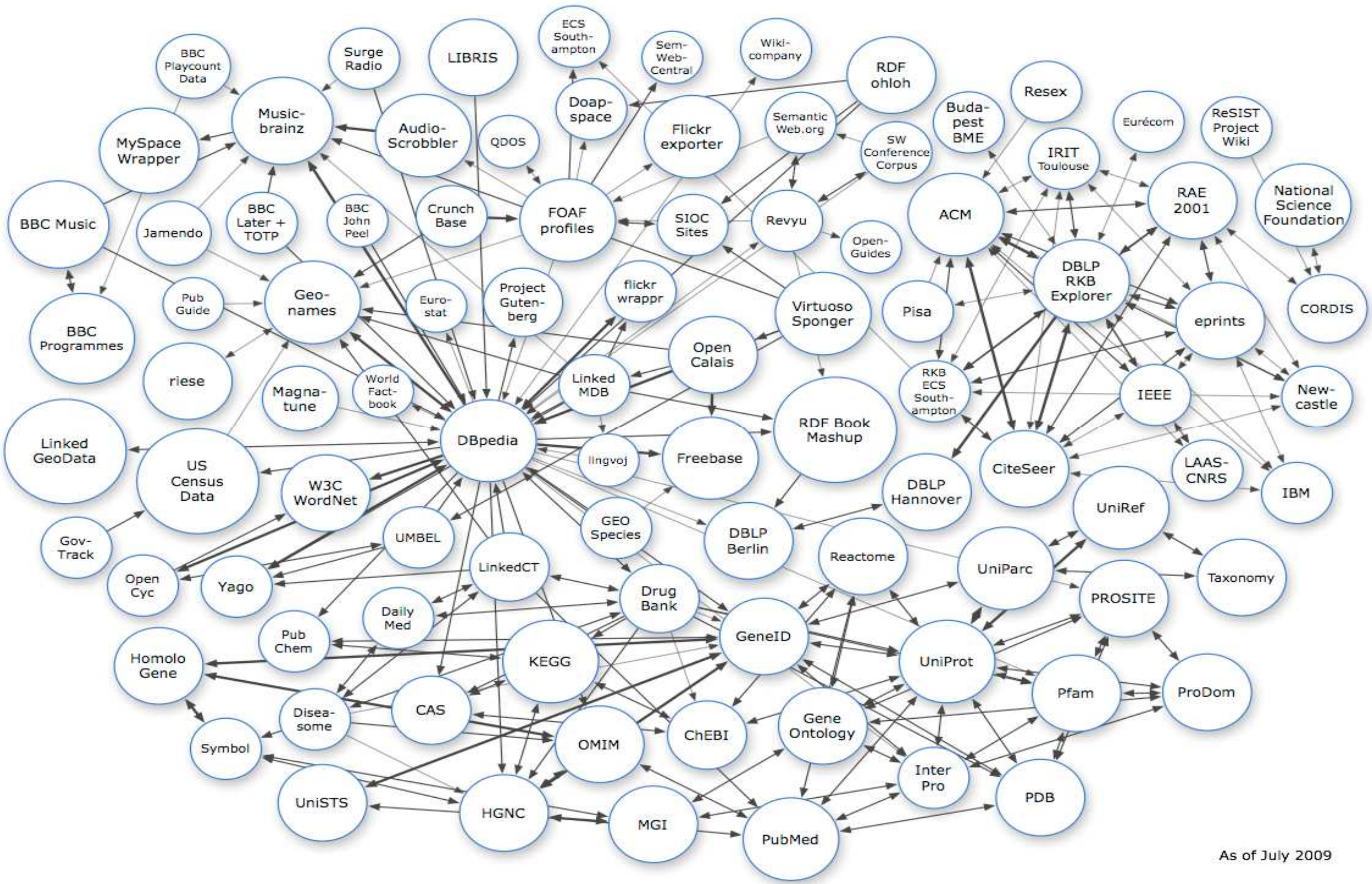
Tim Berners-Lee 2006

1. Use URIs as names for things

1. Use HTTP URIs so that people can look up those names.

1. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)

1. Include links to other URIs. so that they can discover more things.



As of July 2009

Is it really mainstream Semantic Web?

- What is the relationship between the models whose instances are being linked?
- How to do querying on LOD without knowing individual datasets?
- How to perform schema level reasoning over LOD cloud?

Example: GeoNames

Populated Place Features (city, village,...)			
2,518,403	P.PPL	populated place	a city, town, village, or other agglomeration of buildings where people live and work
48,483	P.PPLX	section of populated place	
39,336	P.PPLL	populated locality	an area similar to a locality but with a small group of dwellings or other buildings
13,306	P.PPLQ	abandoned populated place	
2,684	P.PPLA4	seat of a fourth-order administrative division	
2,028	P.PPLA	seat of a first-order administrative division	seat of a first-order administrative division (PPLC takes precedence over PPLA)
1,847	P.PPLW	destroyed populated place	a village, town or city destroyed by a natural disaster, or by war
1,006	P.PPLF	farm village	a populated place where the population is largely engaged in agricultural activities
930	P.PPLA3	seat of a third-order administrative division	
695	P.PPLA2	seat of a second-order administrative division	
253	P.PPLS	populated places	cities, towns, villages, or other agglomerations of buildings where people live and work
249	P.STLMT	israeli settlement	
235	P.PPLC	capital of a political entity	
57	P.		
29	P.PPLR	religious populated place	a populated place whose population is largely engaged in religious occupations
6	P.PPLG	seat of government of a political entity	
2,629,547	Total for P		

Where is the semantics?

Don't get me wrong

Linked Open Data is great, useful, cool, and a very important step.

But if we stay semantics-free, Linked Open Data will be of limited usefulness!

Beyond instance level linkage

- Relationships are at the heart of Semantics.

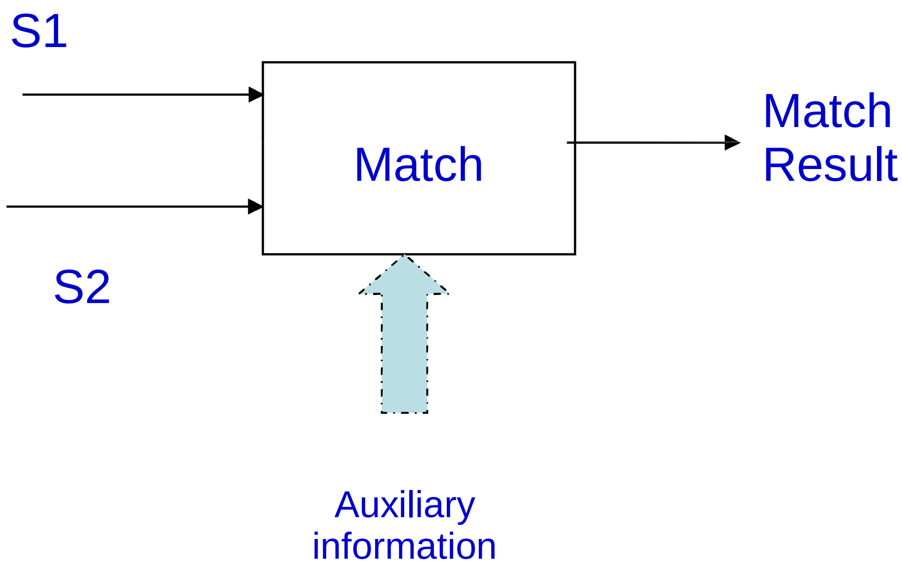
- LOD captures instance level relationships, but lacks class level relationships.
  - Superclass
  - Subclass
  - Equivalence

- How to find these relationships?
  - Perform a matching of the LOD Ontology's using state of the art ontology matching tools.

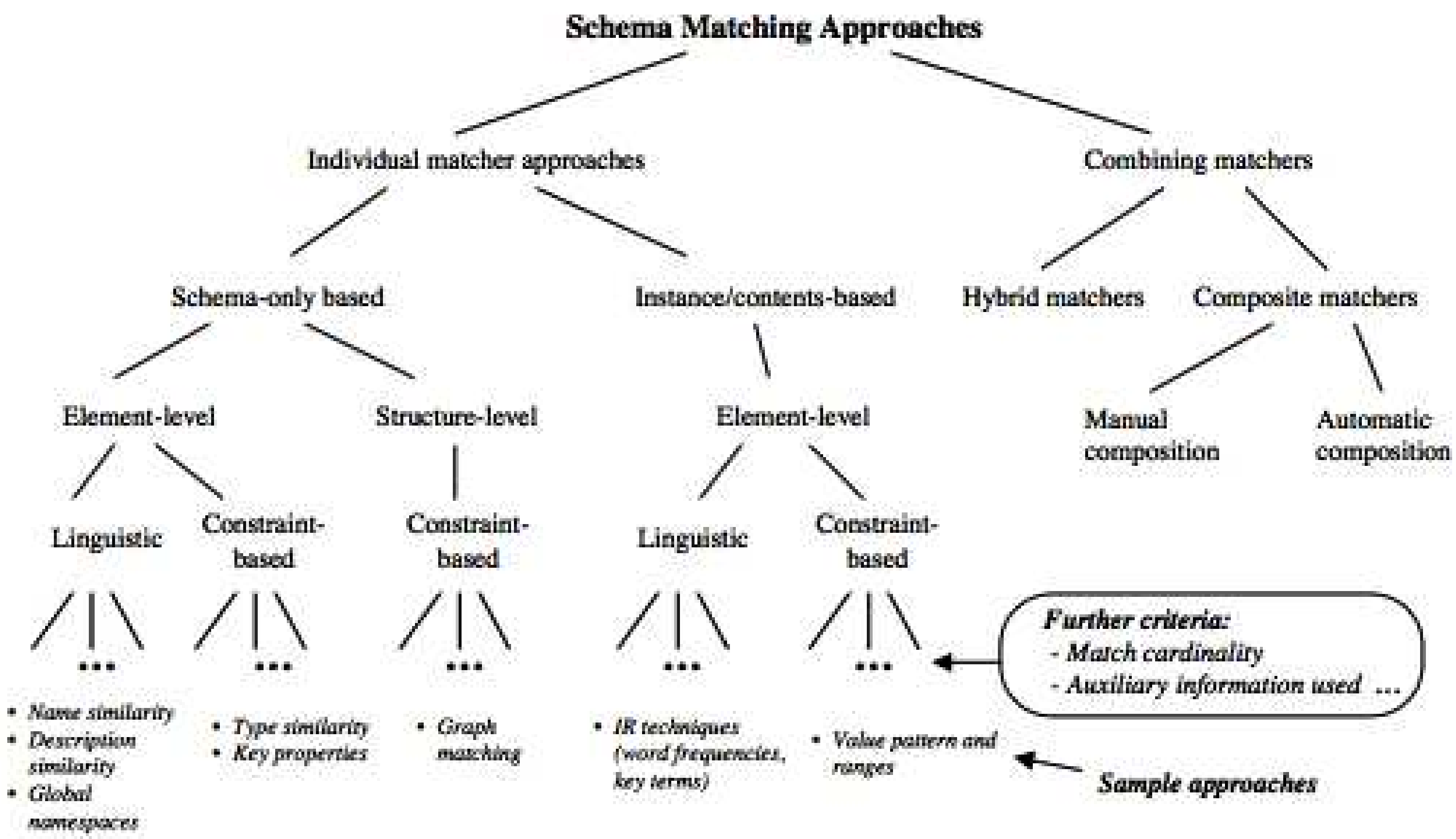
- Desirable
  - Considering the size of LOD, at least have results which a human can create.

Ontology Matching

The task of finding the semantic correspondences between elements of two Ontologys.



Existing Approaches



A survey of approaches to automatic Ontology matching by Erhard Rahm, Philip A. Bernstein in the VLDB Journal 10: 334–350 (2001)

LOD Ontology Alignment

- **Existing systems have difficulty in matching LOD Ontologys!**
  - **Nation = Menstruation, Confidence=0.9 □**

- **They are tuned to perform on the established benchmarks, but not in the wilds.**

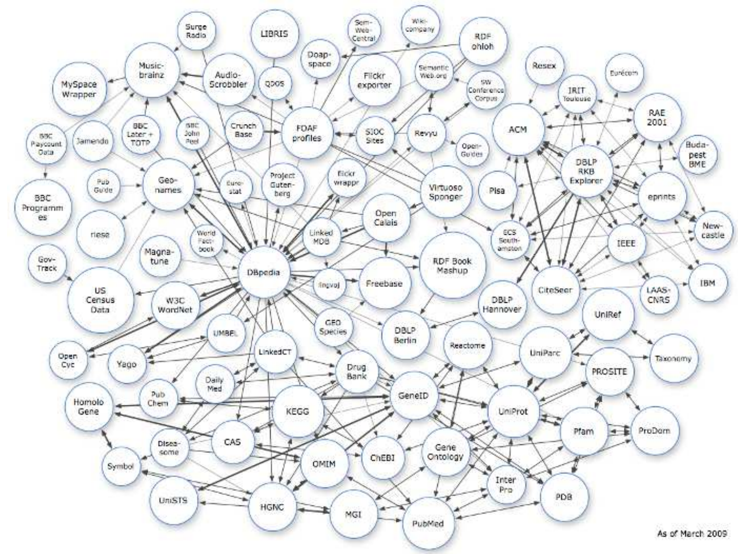
- **LOD Ontology’s are of very different nature**
  - **Created by community for community.**
  - **Emphasis on number of instances, not number of meaningful relationships.**
  - **Require solutions beyond syntactic and structural matching.**

BLOOMS Approach

Use knowledge contributed by users

To improve

knowledge contributed by users



Ontology Matching on LOD using Wikipedia Categorization

- **On Wikipedia, categories are used to organize the entire project.**
- **Wikipedia's category system consists of overlapping trees.**
- **Simple rules for categorization**
  - **“If logical membership of one category implies logical membership of a second, then the first category should be made a subcategory”**
  - **“Pages are not placed directly into every possible category, only into the most specific one in any branch”**
  - **“Every Wikipedia article should belong to at least one category.”**

BLOOMS

1. **Pre-processing of the input ontologies** in order to (i) remove property restrictions, individuals, and properties, and to (ii) tokenize composite class names to obtain a list of all simple words contained within them, with stop words removed.
2. **Construction of the BLOOMS forest** T_C for each class name C, using information from Wikipedia.
3. **Comparison of constructed BLOOMS forests**, which yields decisions which class names are to be aligned.
4. **Post-processing** of the results with the help of the Alignment API and a reasoner.

BLOOMS decisions

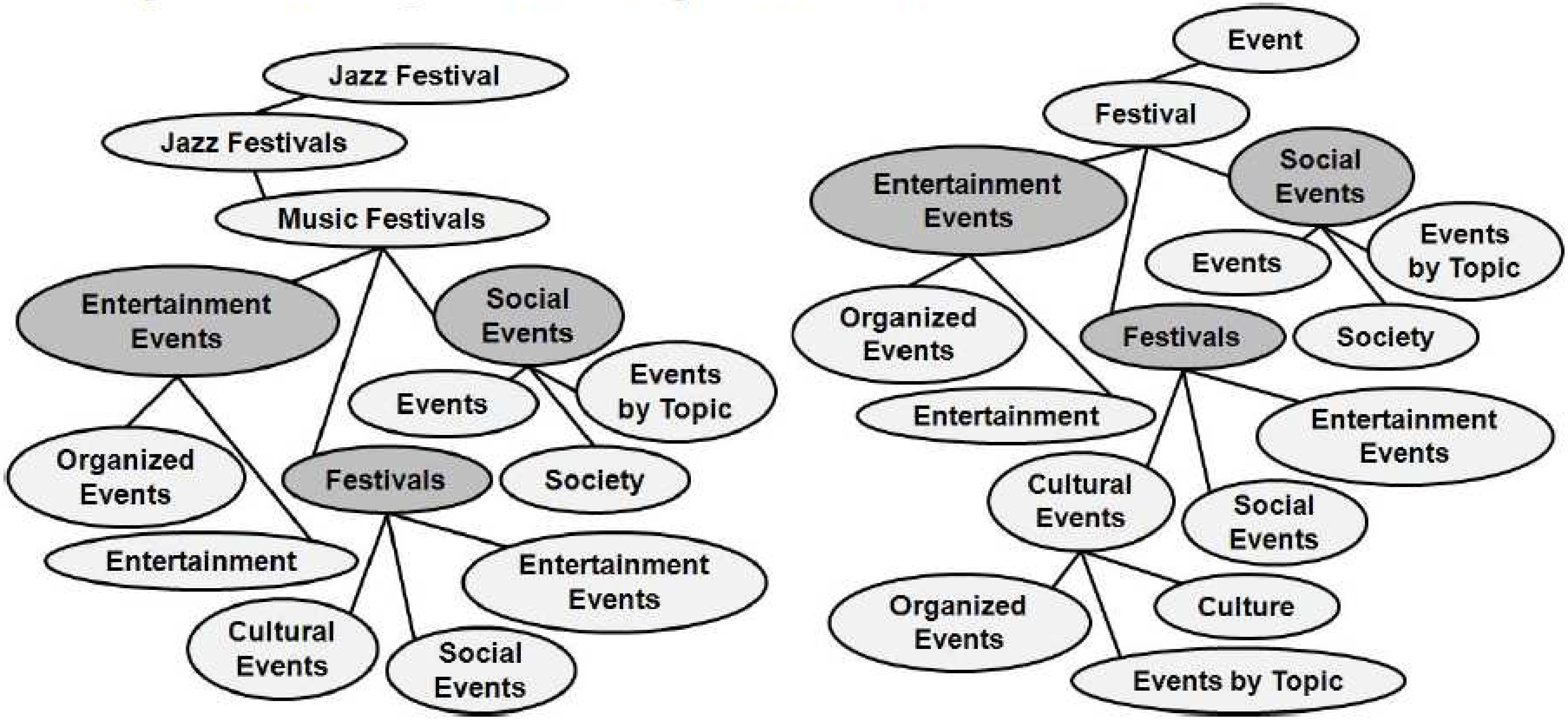
1. Remove from T_s all nodes for which there is a parent node which occurs in T_t. All leaves of the resulting tree T'_s are either of level 4 or occur in T_t. Note that due to the way BLOOMS trees are constructed, we removed only nodes from T_s which actually occur in T_t—we remove them because they do not give us any essential additional information for comparing T_s with T_t.
2. $o(T_s, T_t) = \frac{n}{k-1}$, where n is the number of nodes in T'_s which occur also in T_t, and k is the total number of nodes in T'_s (we do not count the root).

The decision on an alignment is then made as follows.

- If, for any choice of $T_s \in T_C$ and $T_t \in T_D$, we have that $T_s = T_t$, then we set C owl:equivalentClass D.
- If $\min\{o(T_s, T_t), o(T_t, T_s)\} \geq x$ for any choice of $T_s \in T_C$ and $T_t \in T_D$, and for some pre-defined threshold x,[8] then set C rdfs:subClassOf D if $o(T_s, T_t) \leq o(T_t, T_s)$, and set D rdfs:subClassOf C if $o(T_s, T_t) \geq o(T_t, T_s)$.

BLOOMS trees

Fig. 1. BLOOMS trees for Jazz Festival with sense Jazz Festival and for Event with sense Event. To save space, some categories are not expanded to level 4.



Evaluation Objectives

- Examine BLOOMS as a tool for the purpose of LOD Ontology integration.

- Examine the ability of BLOOMS to serve as a general purpose ontology matching system.

BLOOMS

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

BLOOMS

Table 1. Results on the oriented matching track. Results for RiMOM and AROMA have been taken from the OAEI 2009 website. Legends: Prec=Precision, A-API=Alignment API, OMV=OMViaUO, NaN=division by zero, likely due to empty alignment.

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

Thank You!