Cost-Effective Conceptual Design over Taxonomies

Yodsawalai Chodpathumwan

University of Illinois at Urbana-Champaign

Ali Vakilian

Massachusetts Institute of Technology

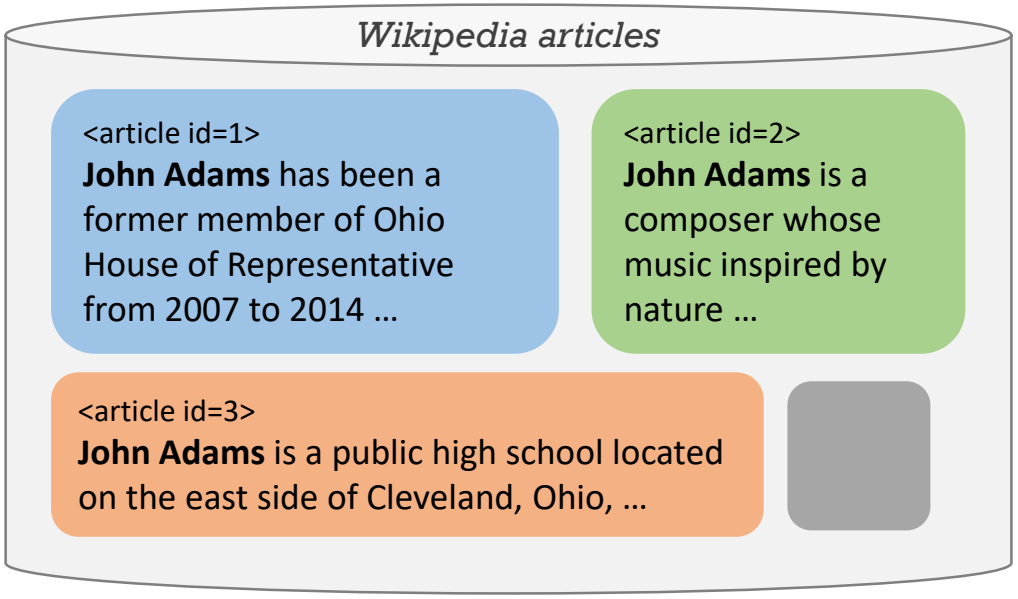
Arash Termehchy, Amir Nayyeri

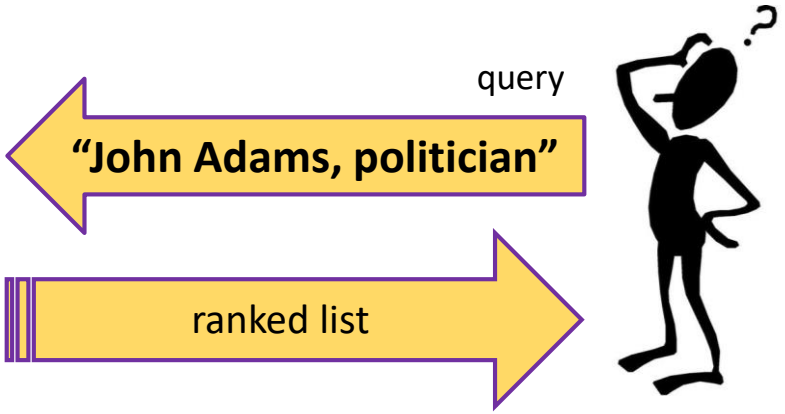
Oregon State University

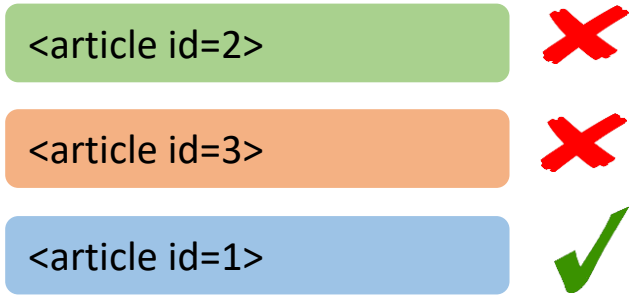
Most information over the web is unstructured.

Medical articles, HTML pages, …

Users have to usually query over unstructured data.







$$\text{Precision@}k = \frac{\#\text{returned relevant answers in top } k \text{ answers}}{\#\text{returned answers in top } k \text{ answers}}$$

precision@3 = **1/3**

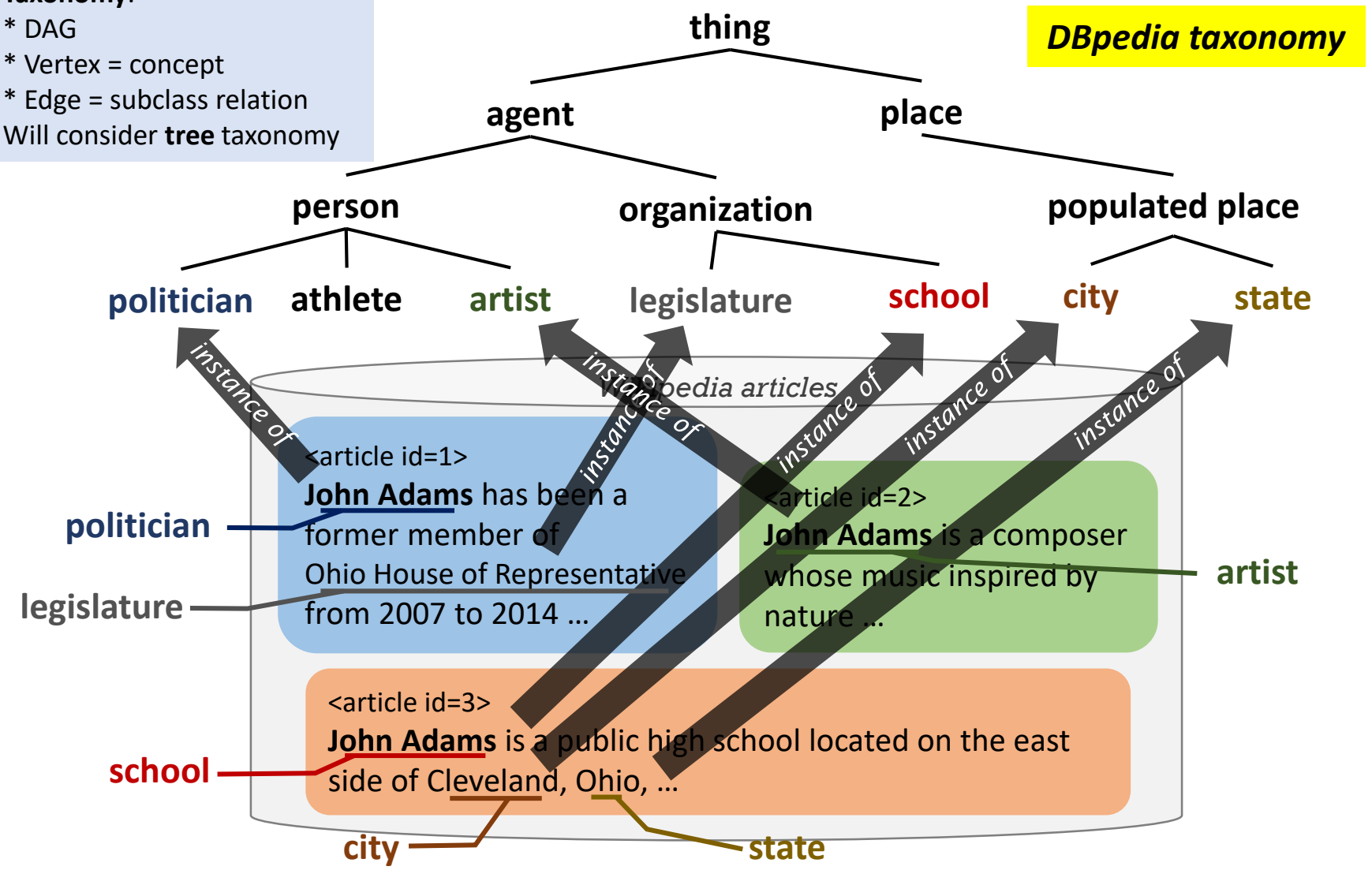
poor ranking quality!

Only Article id **1** is about a politician.

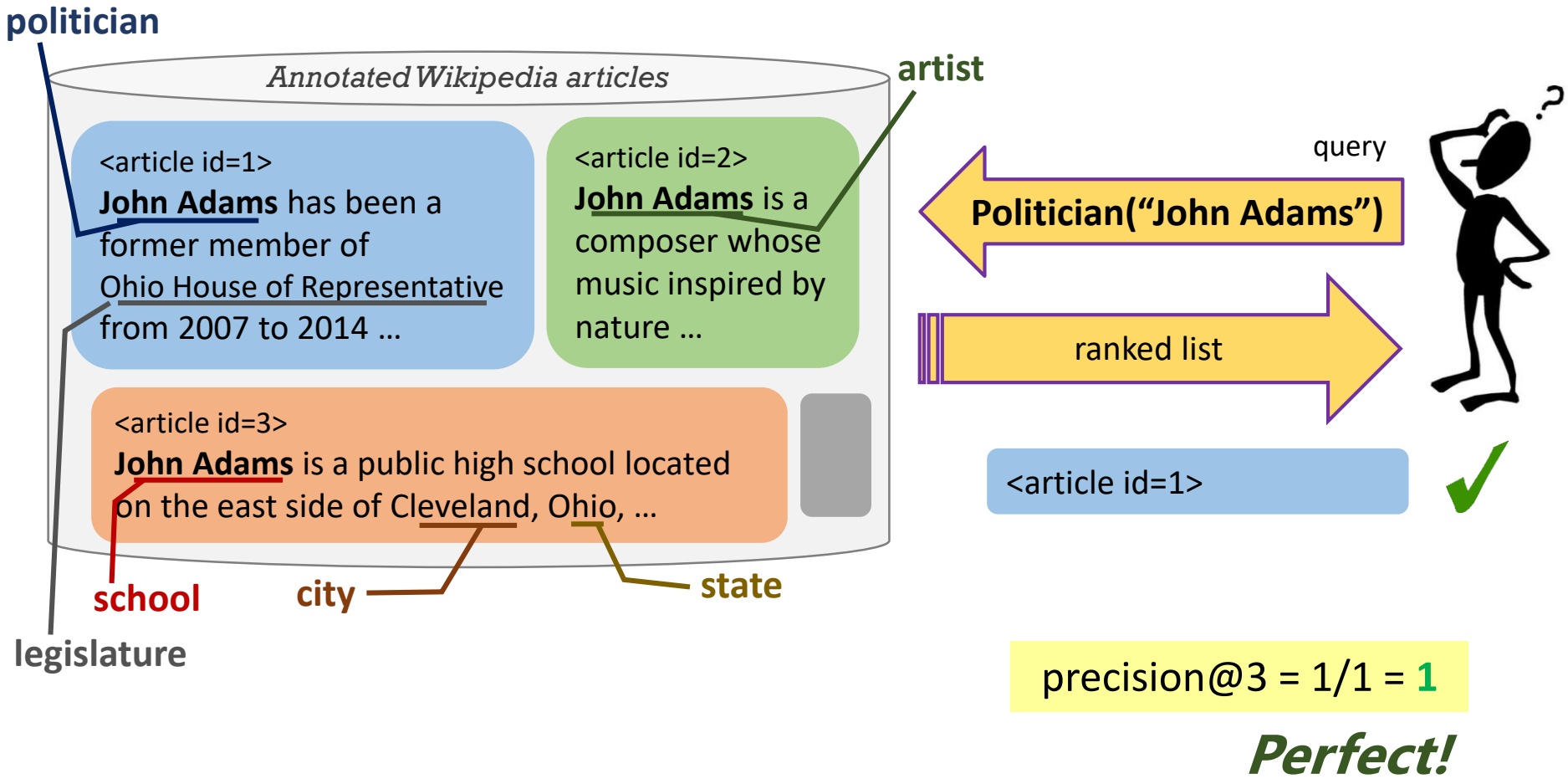
Annotating a dataset improves the effectiveness of answering queries.

Taxonomy:
* DAG
* Vertex = concept
* Edge = subclass relation

Will consider **tree** taxonomy



Users can submit queries with concepts over annotated dataset.



precision@3 = 1/1 = **1**

Perfect!

Concept annotation is costly.

Instances of concepts are annotated by a program called concept annotator.

Researchers estimate that annotating each article in MEDLINE/PubMED dataset using concepts in MeSH taxonomy costs about $9.4 [K.Liu, 2015]**.**

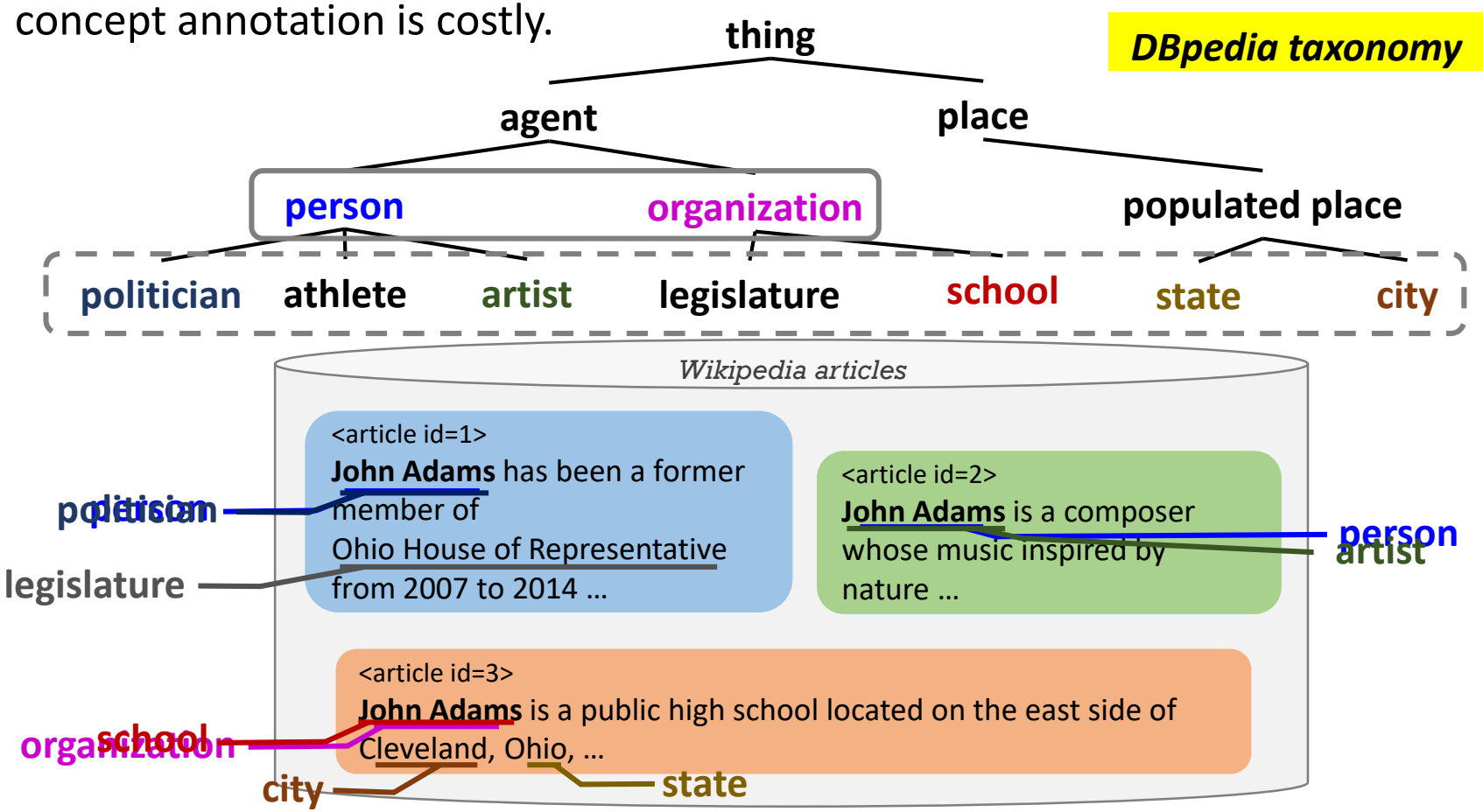
It is costly to develop, execute, and maintain a concept annotator.

- **Development:**
  - Hand-tuned programming rules – need experts, thousands of rules
  - Machine learning technique – find and extract lots of relevant features
- **Execution:** may take several days and require lots of computational resources
- **Maintenance:** datasets evolve over time – rewrite and re-execute concept annotators

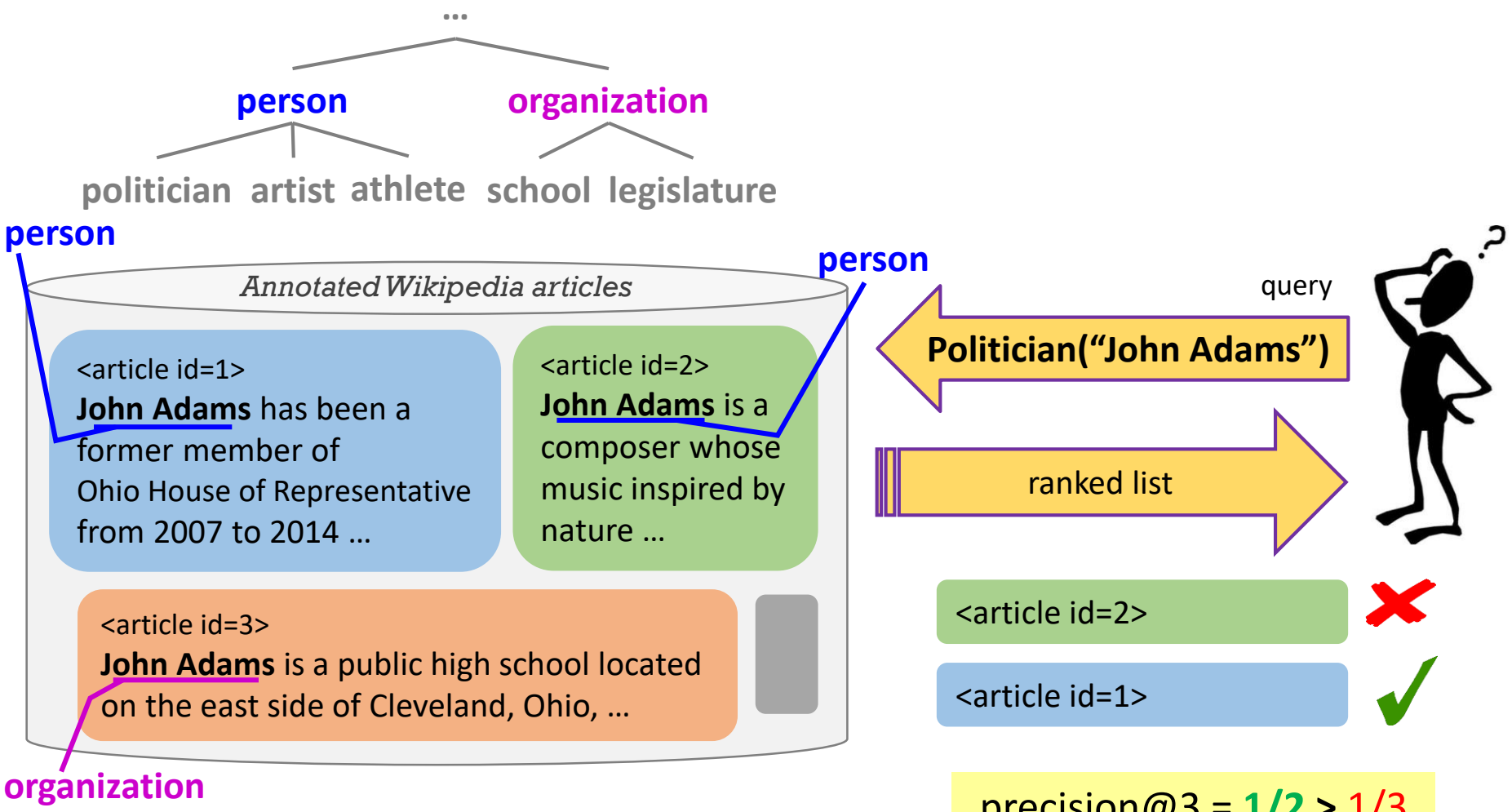
It is not usually possible to annotate all concepts.

Ideally, we would like to annotate instances of all concepts in a given taxonomy from a dataset to answer all queries effectively.

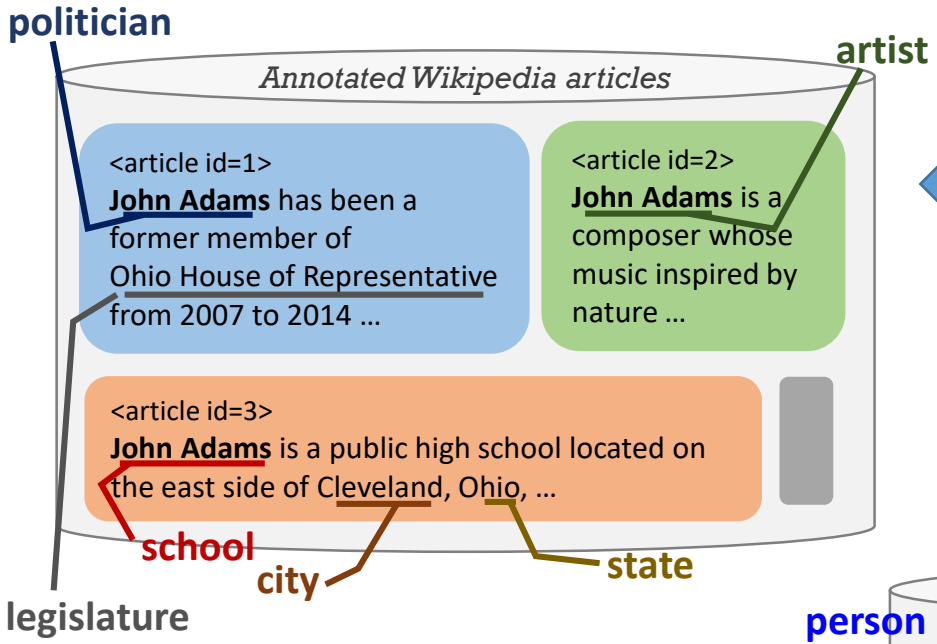
With limited budget, we can only annotate instances of some concepts because concept annotation is costly.



Annotating datasets with only a subset of concepts from a taxonomy still improves the effectiveness of answering queries.

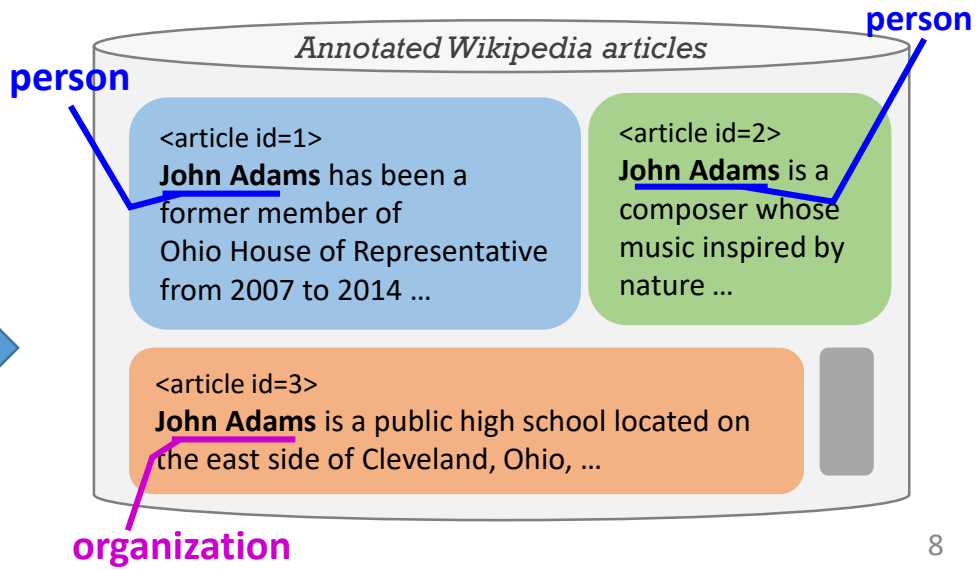


A subset of concepts in a taxonomy used to annotate a dataset is called a conceptual design for the data.



S_1 = {politician, artist, school, city, state, legislature}

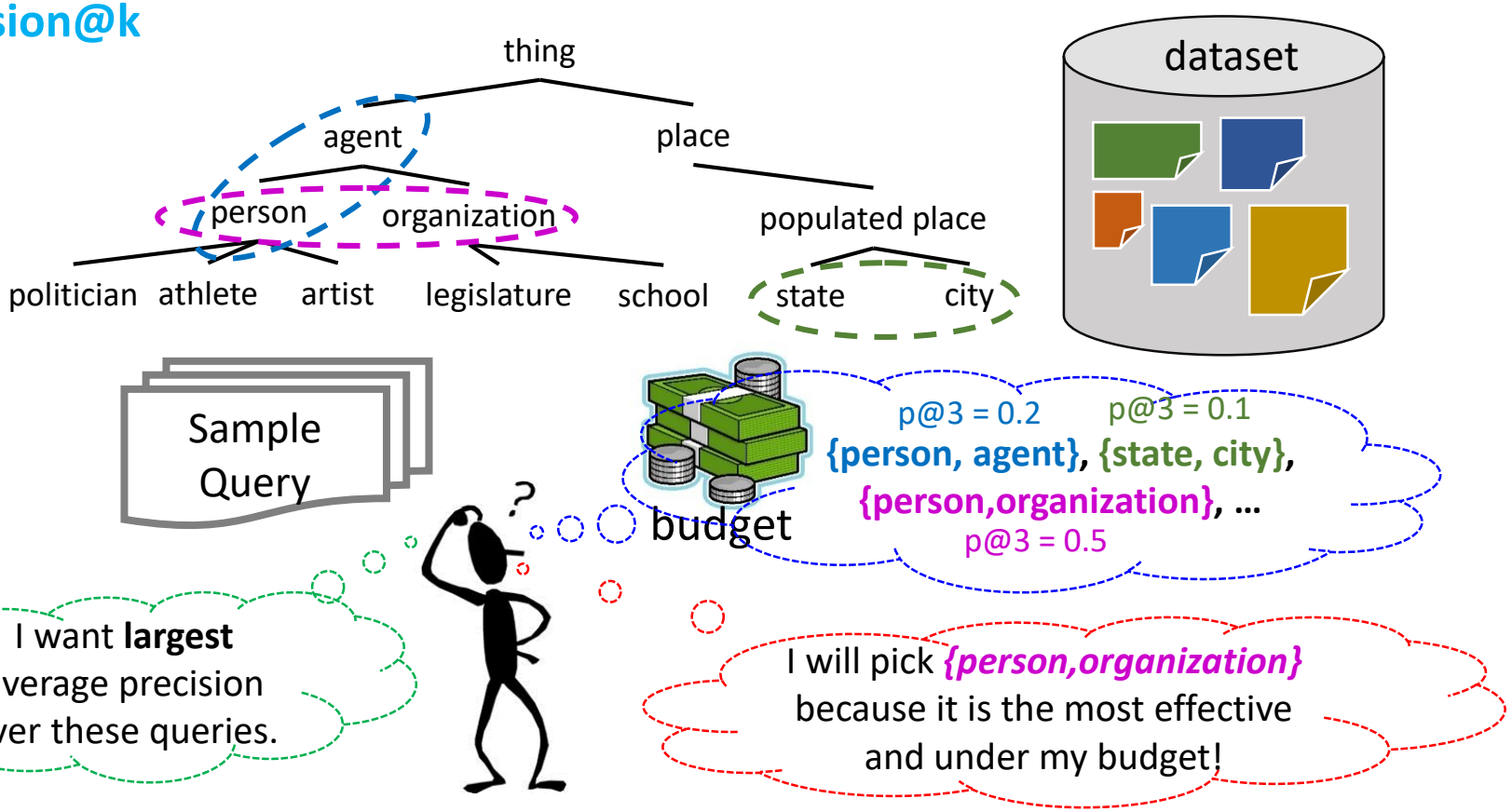
S_2 = {person, organization}



Which conceptual design to pick?

Given a dataset, a taxonomy, a sample of query workload and a budget, find a subset of concepts from an input taxonomy that maximizes the effectiveness of answering queries.

Precision@k



Problem of Cost-Effective Conceptual Design (CECD)

Given a dataset, a sample of query workload, a taxonomy, a available budget

We would like to select a conceptual design S such that

- $\sum_{C \in S} w(C) \leq B$

  Cost function

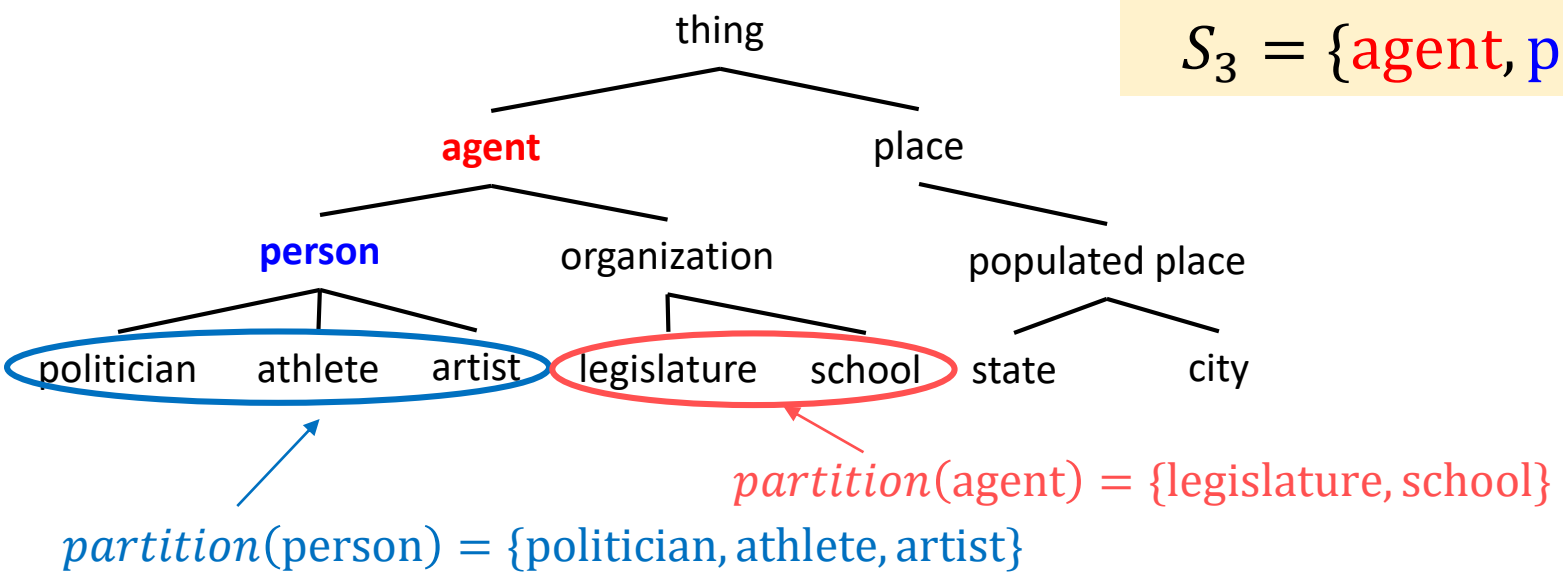
  Budget

- S provides the largest improvement in the average *precision@k* of answering queries amongst all designs that satisfy the budget constraint.

Let's quantify the amount of improvement in precision@k: the **queriability** of a design S or $QU(S)$

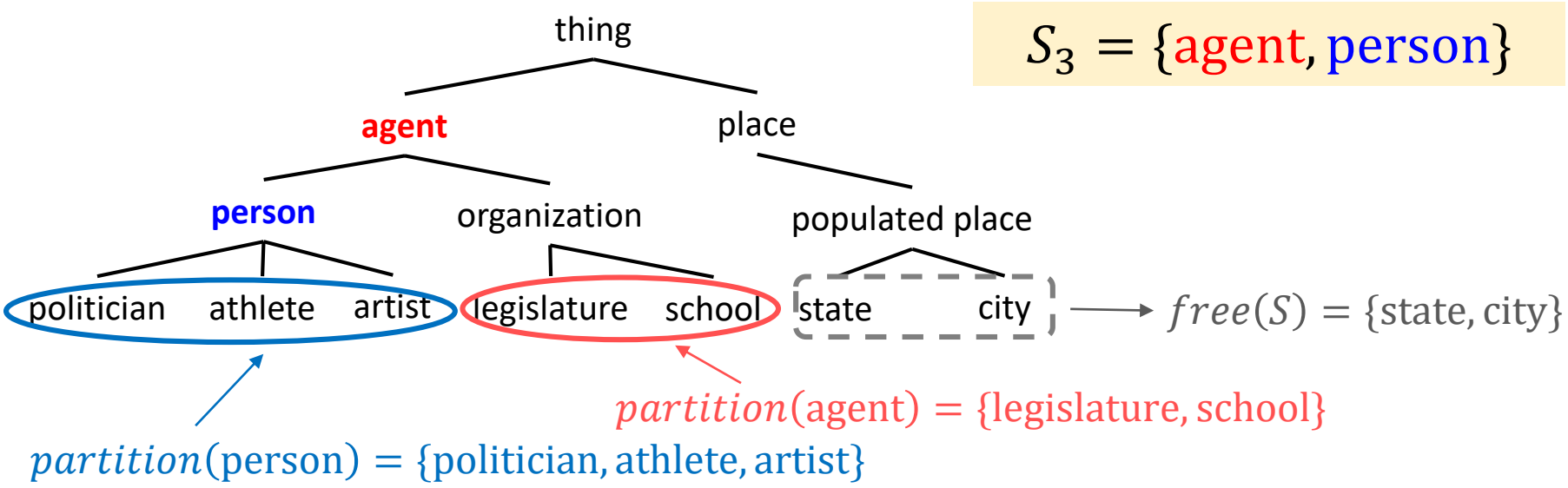
Partitions of a conceptual design

Annotating a concept in a taxonomy also improves quality of answering queries with the concepts that are subclass or descendant of them.



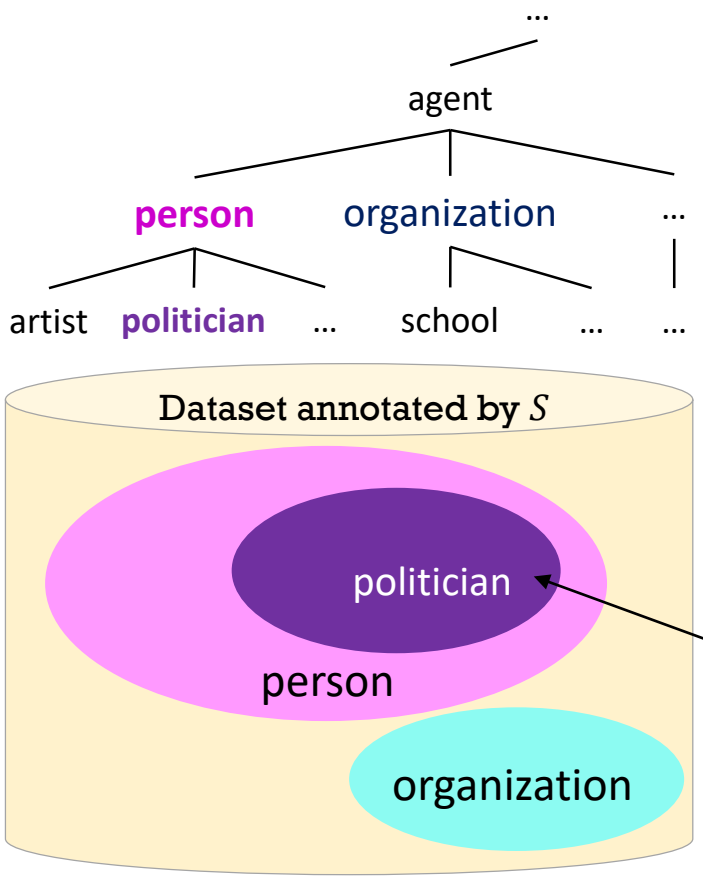
$partition(S)$ is the set of partitions of each concept in the conceptual design S.

A conceptual design may not help all the queries.



A set of leaf concepts that do not belong to any partition of S is called $free(S)$.

Conceptual design S improves the effectiveness of answering queries whose concepts are in ***partitions*** of S.



Query : **Politician("John Adams")**

$S = \{\text{person}, \text{organization}\}$

politician $\in partition(\text{person})$

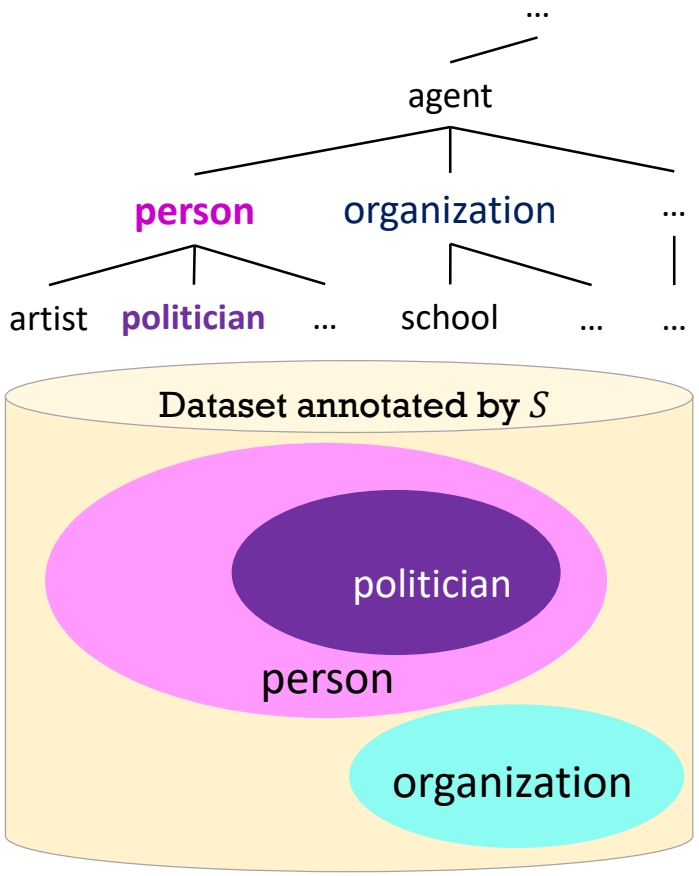
$d(c)$: fraction of documents of concept c in a dataset

Likelihood of returning relevant answers with concept "politician" is

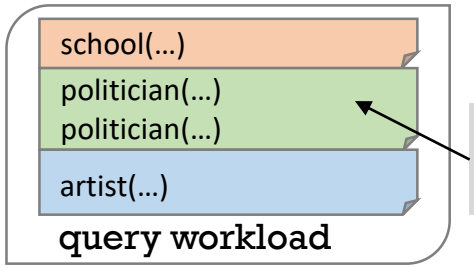
$$\frac{d(\text{politician})}{d(\text{person})}$$

Improvement over unannotated dataset

Conceptual design S improves the effectiveness of answering queries whose concepts are in ***partitions*** of S.



$S = \{\text{person}, \text{organization}\}$



Portion of queries about "politician" is $u(\text{politician})$

Overall improvement for concept "politician" is

$$\frac{u(\text{politician})d(\text{politician})}{d(\text{person})}$$

Total improvement from partition of "person" is

$$\sum_{c \in part(\text{person})} \frac{u(c)d(c)}{d(\text{person})}$$

Total improvement from design S is

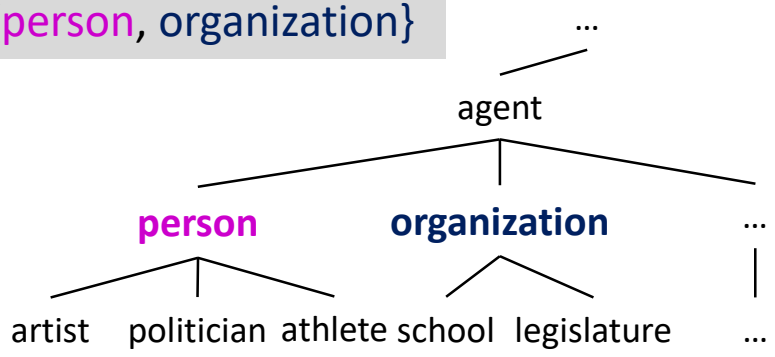
$$\sum_{\boldsymbol{P \in partition(S)}} \sum_{\boldsymbol{c \in partition(P)}} \frac{\boldsymbol{u(c)d(c)}}{\boldsymbol{d(P)}}$$

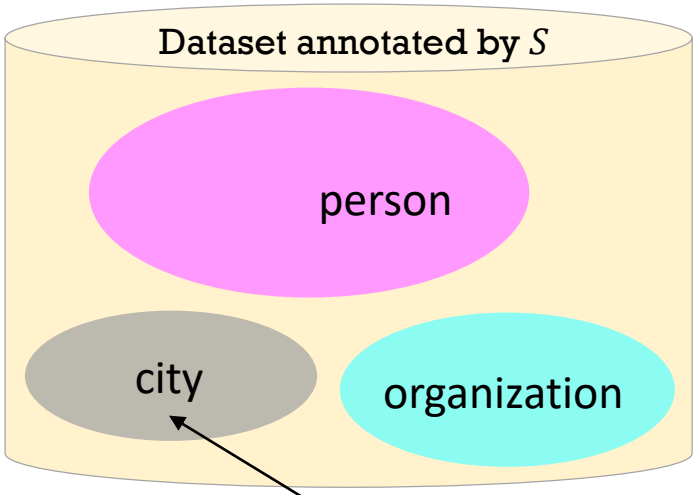
The improvement from a design for queries whose concepts are ***not*** in any partition of the design.

The concepts with more instances in the dataset are more likely to appear in the top answers. Thus, it is more likely they contain some relevant answers for the query.

Query : **City("Washington")**

$S = \{$person, organization$\}$





city $\in free(S)$

The total improvement by concepts in $free(S)$ is $\sum_{c \in free(S)} u(c)d(c)$

Portion of queries whose concepts are c

Portion of documents in the dataset that belong to c

Queriability Function

Given dataset, a query workload and a design S over a taxonomy, the queriability function is

$$QU(S) = \sum_{P \in partition(S)} \sum_{c \in partition(P)} \frac{u(c)d(c)pr(P)}{d(P)} + \sum_{c \in free(S)} u(c)d(c)$$

Formal definition of Cost-Effective Conceptual Design Problem (CECD)

Given a taxonomy X, a dataset D, query workload Q and a budget B, find a **conceptual design** S over X such that

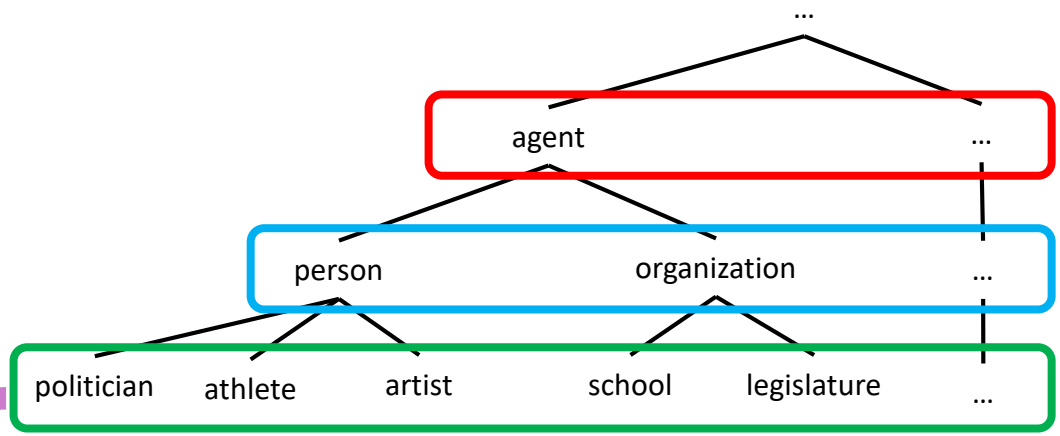
$$\sum_{c \in S} w(c) \leq B$$

and S maximizes the queriablity

$$QU(S) = \sum_{P \in partition(S)} \sum_{c \in partition(P)} \frac{u(c)d(c)pr(P)}{d(P)} + \sum_{c \in free(S)} u(c)d(c)$$

NP-Hard !

We have proposed an approximation algorithm called Level-wise Algorithm (LW)



$QU = APM(\{\text{agent},\ldots\})$

$QU = APM(\{\text{person},\ldots\})$

$QU = APM(\{\text{politician},\ldots\})$

S_{level} ← a *design* with max{QU, QU, QU, … }

S_{leaf} ← leaf concept with largest popularity (u)

APM algorithm returns a design with largest queriability over a set of concepts.

[Termehchy, SIGMOD'14]

Return a design with $\max\{QU(S_{level}), QU(S_{leaf})\}$

Level-wise algorithm has a bounded approximation ratio over a special case of the CECD problem

- Sometimes it is easier to use and manage a conceptual design whose concepts are not subclass/superclass of each other.
  - We call this design a **disjoint design**.

- May restrict the solution in the CECD problem to disjoint designs.
  - We call this problem a **disjoint CECD problem**.

Theorem
The Level-wise algorithm is a $O(\log |C|)$-approximation for the disjoint CECD problem.

Experiment Settings

- 8 extracted tree taxonomies from YAGO ontology, T1-T8
  - Number of concepts between 10 – 400 with height of 2 – 9
- Datasets of articles from English Wikipedia Collection
  - ~1.5 million articles
- Subset of Bing *(bing.com)* query log whose relevant answers are Wikipedia article.
  - ~4000 queries
- Effectiveness metric: precision at 3 ($p@3$)
- Two cost models: uniform cost and random cost

Accuracy of Queriability Function

- **Oracle**: enumerates all feasible designs and selects a design with maximum precision at 3.
- **Queriability Maximization** (QM): enumerates all feasible designs and selects a design with maximum queriability.

B=1 : enough budget to annotate all concepts

	B	T1		T2		T3	
		Oracle	**QM**	Oracle	**QM**	Oracle	**QM**
Uniform	0.1	0.149	0.149	0.241	0.232	0.222	0.210
	0.2	0.168	0.168	**0.303**	0.285	**0.281**	0.269
	0.3	0.177	0.177	0.318	0.315	0.304	0.304
	0.4	0.192	0.192	0.320	0.318	0.306	0.304
	0.5	0.193	0.193	0.326	0.324	0.306	0.306
	0.6	0.195	0.195	0.326	0.326	0.306	0.306
	0.7	0.195	0.195	0.326	0.326	0.306	0.306
	0.8	0.195	0.195	0.326	0.326	0.306	0.306
	0.9	0.195	0.195	0.316	0.316	0.306	0.306

Results for random cost is similar to the results for uniform cost

Level-wise algorithm is effective.

- Compare LW with APM.

B=1 : enough budget to annotate all concepts

	B	T1		.T2		T3		T4		T5		T6		T7		T8	
		APM	**LW**	APM	**LW**	APM	**LW**	APM	**LW**	APM	**LW**	APM	**LW**	APM	**LW**	APM	**LW**
Uniform	0.1	.089	**.103**	.234	.232	.208	.210	.158	**.179**	.178	**.206**	.229	**.240**	.243	**.254**	.250	**.259**
	0.2	.149	**.164**	.253	.285	.258	.269	.177	**.212**	.214	**.227**	.244	**.248**	.259	**.261**	.262	.263
	0.3	.164	.164	.292	**.316**	.288	.297	.191	**.231**	.228	**.242**	.247	.248	.260	.261	.262	.263
	0.4	.164	**.183**	.320	.318	.297	**.304**	.215	**.240**	.237	**.248**	.248	.248	.261	.261	.263	.263
	0.5	.183	**.192**	.323	.323	.304	.306	.229	**.241**	.241	**.250**	.248	.248	.261	.261	.263	.263
	0.6	.192	.194	.323	.323	.304	.306	.229	**.241**	.249	.250	.248	.248	.261	.261	.263	.263
	0.7	.193	.195	.323	.323	.306	.306	.235	.241	.249	.250	.248	.248	.261	.261	.263	.263
	0.8	.195	.195	.323	.323	.306	.306	.239	.241	.249	.250	.248	.248	.261	.261	.263	.263
	0.9	.195	.195	.323	.323	.306	.306	.241	.241	.250	.250	.248	.248	.261	.261	.263	.263

Results for random cost is similar to the results for uniform cost

Level-wise algorithm is efficient.

Time in seconds	T4	T5	T6	T7	T8
LW	2	2	5	6	6
APM	2	2	2	13	40
Size of taxonomy	28	63	185	279	387

Conclusion & On-going Work

- We introduced the cost-effective conceptual design over taxonomies
- We proposed an efficient approximation (LW) algorithm for the problem.
- Our empirical results showed that LW is generally effective and scalable.

We are working on variations of the problem including taxonomies that are directed acyclic graphs and queries that refer to multiple concepts.

More info in our technical report (arXiv:1503.05656)