• Header			Interaction count:
Select * { ?x lives_in columbia h }	india . nas_graduate ?x	Select * { ?x <livesin> ?x <graduat ?s2 <dealsw }</dealsw </graduat </livesin>	?s2 . edFrom> <columbia_university> . /ith&gt; <india></india></columbia_university>
feedback	original_element	proposed_element	assignment_example
	?x	?x	<fazlollah_reza></fazlollah_reza>
	lives_in	<li>livesIn&gt;</li>	
$\mathbf{X}$		?s2	<canada></canada>
	less graduate	canadicate of Frances	

# **Provenance-Based SPARQL Query Formulation**





Department of Computer Science Faculty of Exact Sciences Bar-Ilan University

## Querying a Knowledge Graph



## Querying a Knowledge Graph

Find individuals who are graduates of Columbia University and who now live in India



Issue 1: getting "stuck". What if the result is empty? Residence location is typically a *city*  **Issue 2:** sparsity and heterogeneity

How can we find relevant examples?

#### **Semi-formal Queries**

Find individuals who are graduates of Columbia University and who now live in India Select \* {
 columbia graduate ?x
 ?x lives\_in india

#### Natural Language query: Unstructured, not grounded

Easy to write 🙂

Difficult to ground  $\ensuremath{\mathfrak{S}}$ 

#### Semi-formal query:

Well-structured, not grounded

Requires SPARQL familiarity

#### Easier to ground

Select \* {

?x <graduatedFrom> <Columbia\_University>
?x <lives\_in> ?s2
?s2 <isLocatedIn> <India>

#### Formal query:

Well-structured, grounded

Requires familiarity with SPARQL and KG ☺

Can be directly evaluated  $\bigcirc$ 

### The SPARQLIt Framework: Highlights

#### Provenance-Based SPARQL Query Formulation





Facts as triples:

lives in India Yael India. created on 1947 ∈Pred ∈Ent ∈Ent/Lit Basic graph patterns (BGPs): **?**x lives in ?y. **?**y created on 1947 Query evaluation:  $\varphi_1$ : **?x** $\mapsto$ Yael, **?y** $\mapsto$ India Variable bindings  $\varphi_2$ : **?x** $\mapsto$ Kareena Kapoor, **?y** $\mapsto$ India ... Q(G)={(Yael, India), (Kareena\_Kapoor, India), ...) Query answer

#### User inputs a **semi-formal** query



#### **Step 1: Structural Edits**





### **Structural Edits Provenance**

Given two (semi/formal) BGPs Q, Q'their transformation provenance is prov(Q, Q') = (P, C)





#### **Step 2: Grounding Generator**



### **Step 3: Triple Store**



## **Step 4: Feedback**

Header

#### Interaction count: 1

Select * {	<b>m</b>		Select * {	5	-	ē	0
?x lives_in india .	*	•••	?x <livesin> <india> .</india></livesin>			·⊞	
columbia has_graduate ?x			?x <graduatedfrom> <c< td=""><td>Columb</td><td>ia_U</td><td>nivers</td><td>ity&gt;</td></c<></graduatedfrom>	Columb	ia_U	nivers	ity>
}			}				

feedback	original_element	proposed_element	assignment_example
	?x	?x	<prakash_apte></prakash_apte>
	lives_in	<livesin></livesin>	
	india	<india></india>	
	has_graduate	<graduatedfrom></graduatedfrom>	
	columbia	<columbia_university></columbia_university>	

# **Step 4: Feedback**

• Header			Interaction count: 2		
Select * { ?x lives_in columbia ł }	india . nas_graduate ?x	Select * { ?x <livesin> ?s2 ?x <graduatedf ?s2 <dealswith }</dealswith </graduatedf </livesin>	Select * { ?x <livesin> ?s2 . ?x <graduatedfrom> <columbia_university> . ?s2 <dealswith> <india> }</india></dealswith></columbia_university></graduatedfrom></livesin>		
feedback	original_element	proposed_element	assignment_example		
	?x	?x	<fazlollah_reza></fazlollah_reza>		
	lives_in	<livesin></livesin>			
$\mathbf{\times}$		?s2	<canada></canada>		
	has_graduate	<graduatedfrom></graduatedfrom>			
	columbia	<columbia_university></columbia_university>			
		<dealswith></dealswith>			



### **Experimental Setup**

- Datasets
  - YAGO
  - DBpedia
- Queries: QALD-9 (NL to formal in DBpedia)
- Semi-formal queries:
  - Manual translation
  - YAGO vs DBpedia
- Baselines
  - gAnswer
  - Restricted variants

#### **Some Results**

	SPARQLIt+ QALD Translated	SPARQLIt+ QALD Cross-KB	With-Empty+ QALD Translated	gAnswer+ QALD-9
DBpedia	78%	81%	64%	36%
YAGO	84%	81%	64%	N/A



#### **Some Results**



## Summary

- An interactive framework for SPARQL query construction
  - Using provenance to track query transformations and output
- Search engine + triple store
   + fine-grained user feedback
   help pruning candidates

Header Interaction count: Select \* { Select \* { n ?x lives\_in india. ?x <livesIn> ?s2. columbia has graduate ?x ?x <graduatedFrom> <Columbia\_University>. ?s2 <dealsWith> <India> original element proposed\_element feedback assignment example <Fazlollah Reza> ?x ?x lives in <livesIn> X ?s2 <Canada>

- Future work:
  - Extending query fragment (to e.g., aggregation)
  - Additional operations

