Provenance for Natural Language Queries

Daniel Deutch, Nave Frost, Amir Gilad: Provenance for Natural Language Queries. PVLDB 10(5): 577-588 (2017)

Presented by:

Deep Narayan Mishra

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Outline

- Introduction to NL DB interface
- Why Prov. for NL queries
- Key ideas
- NaLIR Overview
- Basic Solution
- Factorization
- Summarization
- Conclusion
- Limitations

Natural language interface for Databases

- Allows user to access information stored in a database by formulating request in Natural language
- Examples:

NL Queries:

Show me a list of all the customers in my database.
Are there any customers?
List all customers.
Customers.

SQL:

Select customer_name **from** Customers

Result:

Customer names are John. V., Tom, … , etc..

Benefits of NL interface

- Easy access to information stored
- Greater and global accessibility
- Accurate retrieval
- Improved customer service

Why provenance for NL queries?

User Query

return the organization of authors who published papers in database conference after 2005

Formal Query - Conjunctive Query

query(oname) :- **org**(oid,oname), **conf**(cid,cname),
pub(wid,cid,ptitle,pyear), **author**(aid,aname,oid),
domainConf(cid,did), **domain**(did,dname),
writes(aid,wid), **dname**='Database' **pyear**>2005

Result

The organization name is Tel Aviv University (TAU)

How?

Why provenance for NL queries?

What we have – Provenance

(oname,TAU) . (aname,Tova M.) . (ptitle,OASSIS...) . (cname,SIGMOD) . (pyear,14')

Why provenance for NL queries?

User Query

return the organization of authors who published papers in database conference after 2005

Formal Query - Conjunctive Query

query(oname) :- **org**(oid,oname), **conf**(cid,cname),
pub(wid,cid,ptitle,pyear), **author**(aid,aname,oid),
domainConf(cid,did), **domain**(did,dname),
writes(aid,wid), **dname**='Databases', **pyear**>2005

What we want – Explanations

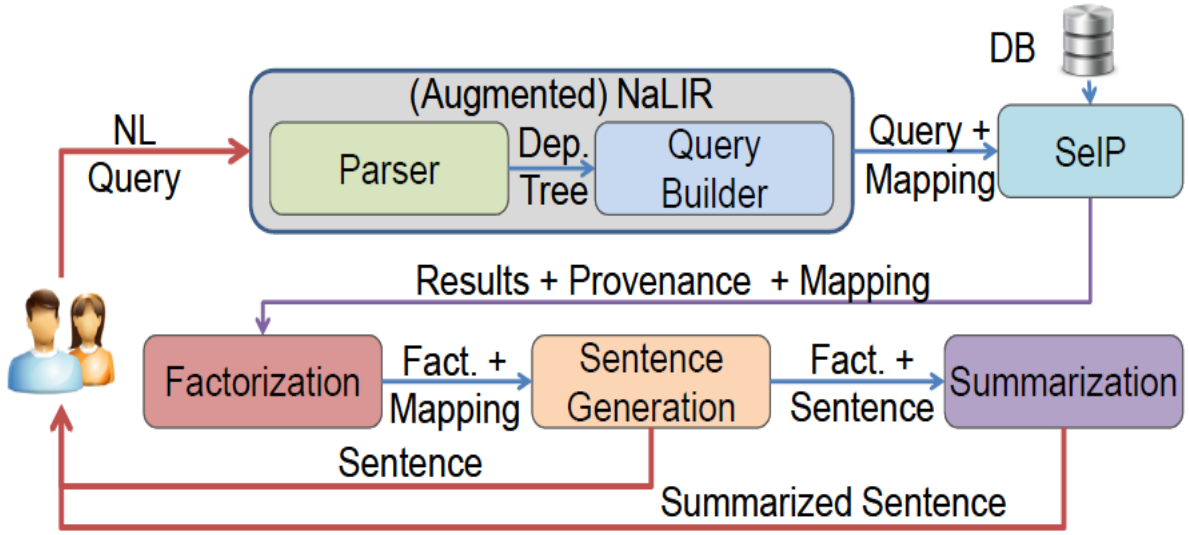
TAU is the organization of Tova M. who published 'OASSIS...' in SIGMOD in 2014

Key Ideas For Generating Provenance

- Provenance tracking using NL query structure
  - Leverage NaLIR system
  - Evaluate formal query using provenance aware engine (SelP).
  - Compose and translate provenance to NL text
- Factorization of provenance
  - Group assignments
  - Looks for compatible structure
- Summarization of provenance
  - Replaces details by their synopsis

Framework of NLProv

- Augment NaLIR
- Use a provenance-aware engine – SelP
- Store the provenance and mappings
- Translate result and provenance to NL using factorization and summarization

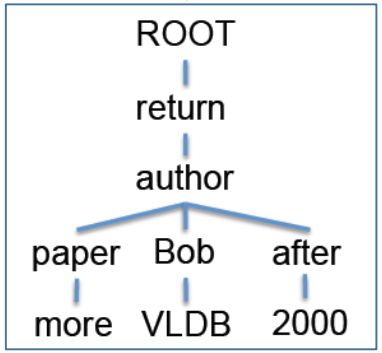


NL to Formal Queries

Overview of NaLIR

- An interactive NL query interface for RDBMS
- Design:
  - Parsing
  - Match + Intermediate representation
  - Query generation

return authors who have more paper than Bob in VLDB after 2000

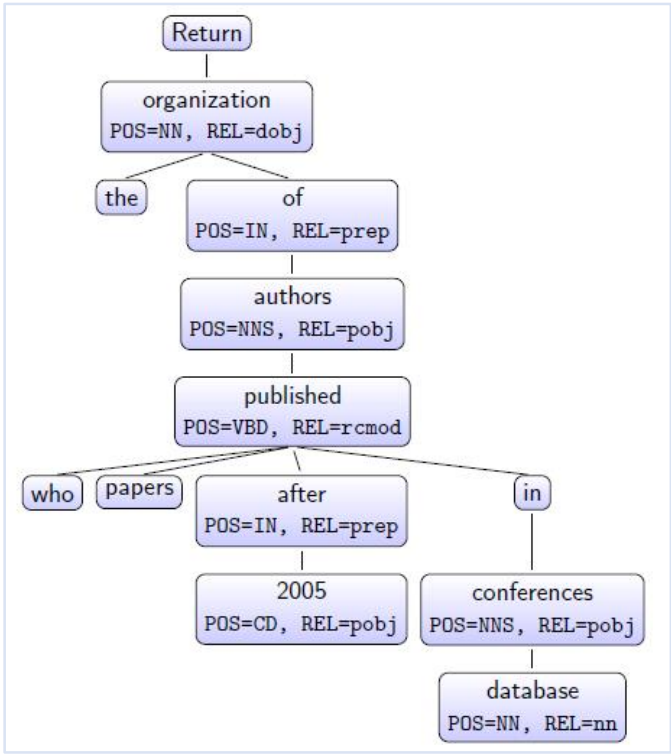


SELECT AUTHOR_NAME FORM AUTHORS
WHERE ...

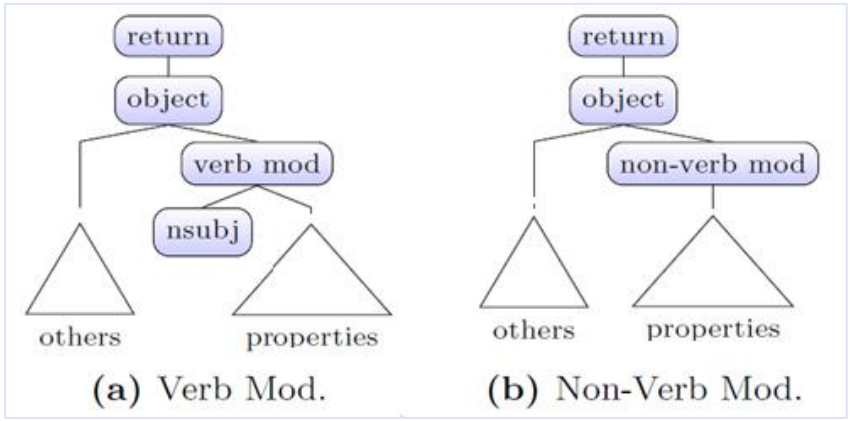
NL to Formal Queries

Dependency Tree

- Every node in dependency tree labeled as:
  - POS (Part of Speech)
  - REL (Relationship with parent node)



- NL query follow one of the two general forms:
  - Sub-sentence rooted by verb modifier
  - Sub-sentence rooted by non-verb modifier



NL to Formal Queries

Conjunctive Query:

- Most common form of query; equivalent to select-project-join (SPJ) queries.
- Easy to analyze.
- Form: $\mathbf{q(x)\ :-\ p_1(x_1),\ p_2(x_2)\ ,\ldots..,\ p_n(x_n)}$
- Example:

SQL: **SELECT** $t_1.a_1,\ldots, t_k.a_k$

FROM R_1 as $t_1,\ldots, R_n$ as t_n

WHERE C

CQ: $q(t_1.a_1,\ldots, t_k.a_k)$:- $R_1(t_1),\ldots, R_n(t_n)$, C

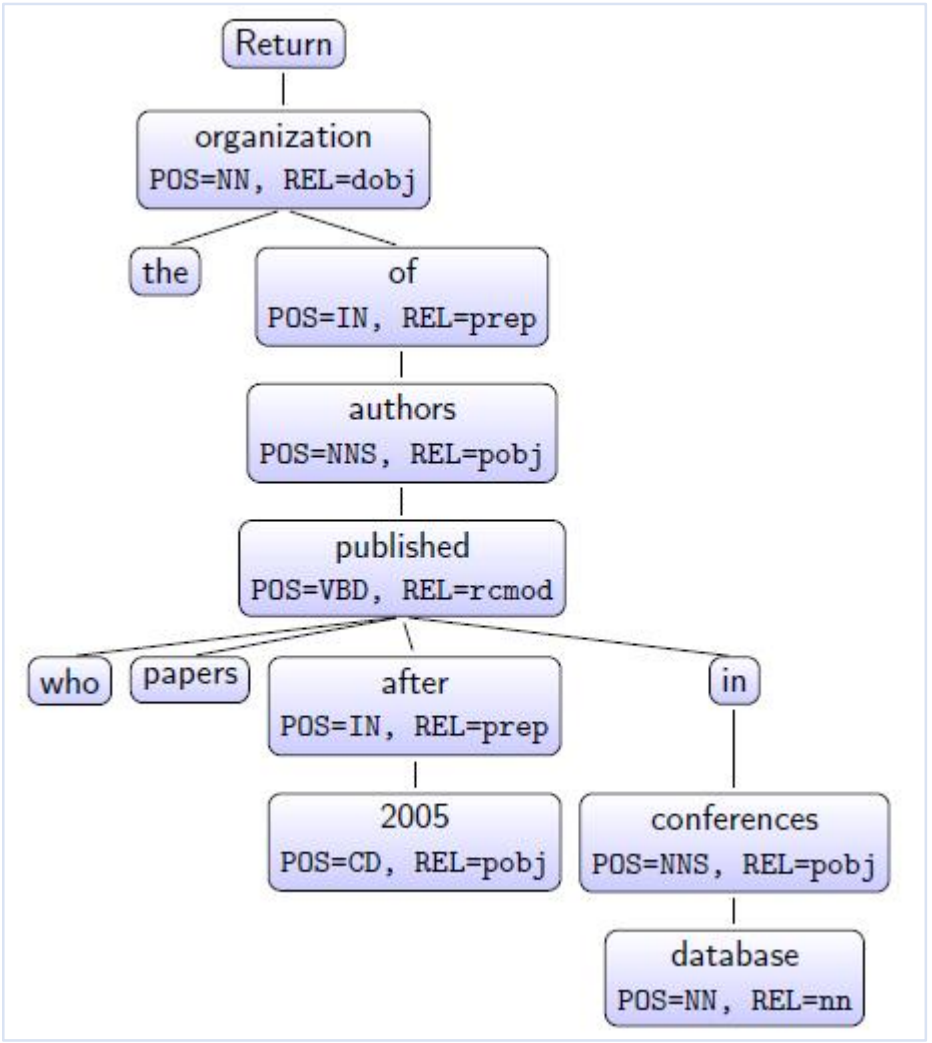
NL to Formal Queries - Example

return the organization of authors who published papers in database conference after 2005

NL Query

query(oname) :- **org**(oid, oname), **conf**(cid, cname),
pub(wid, cid, ptitle, pyear),
author(aid, aname, oid), **domainConf**(cid, did),
domain(did, dname), **writes**(aid, wid),
dname = 'Databases', **pyear** > 2005

CQ Q



Dependency Query Tree

Basic Solution

Store mappings from NL query processing

- Dependency-to-Query-Mapping

  Dependency tree T = (V, E, L) is mapped to CQ Q

$$\tau : V \rightarrow Vars(Q)$$

- Assignments of query variables to DB values

  Atom $R(x_1,\ldots,x_n)$ mapped to tuple $R(a_1,\ldots a_n)$

$$\alpha(x_i) = a_i$$

- Value-Level Provenance:

  A(Q, D) Set of assignments for a CQ Q and a database instance D

$$\sum_{\alpha \in A(Q,D)} \Pi_{\{x_i, a_i | \alpha(x_i) = a_i\}}(x_i, a_i)$$

Mappings - Example

```
query(oname) :- org(oid, oname), conf(cid, cname),
                pub(wid, cid, ptitle, pyear),
                author(aid, aname, oid), domainConf(cid, did),
                domain(did, dname), writes(aid, wid),
                dname = 'Databases', pyear > 2005
```

CQ Q

Rel. *org*

oid	oname
1	UPENN
2	TAU

Rel. *author*

aid	aname	oid
3	Susan D.	1
4	Tova M.	2
5	Slava N.	2

Rel. *writes*

aid	wid
4	6
3	6
5	6
4	7
4	8
4	9

Rel. *pub*

wid	cid	ptitle	pyear
6	10	"OASSIS..."	2014
7	10	"A sample..."	2014
8	11	"Monitoring..."	2007
9	11	"Querying..."	2006

Rel. *conf*

cid	cname
10	SIGMOD
11	VLDB

Rel. *domainConf*

cid	did
10	18
11	18

Rel. *domain*

did	name
18	Databases

Database Instance

(oname,TAU).(aname,Tova M.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')+
(oname,TAU).(aname,Tova M.). (ptitle,Querying...).(cname,VLDB).(pyear,06')+
(oname,TAU).(aname,Tova M.). (ptitle,Monitoring..).(cname,VLDB).(pyear,07')+
(oname,TAU).(aname,Slava N.).(ptitle,OASSIS...). (cname,SIGMOD)(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,A sample...). (cname,SIGMOD).(pyear,14')+
(oname,UPENN).(aname,Susan D.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')

Value-level Provenance

Compute Answer Tree

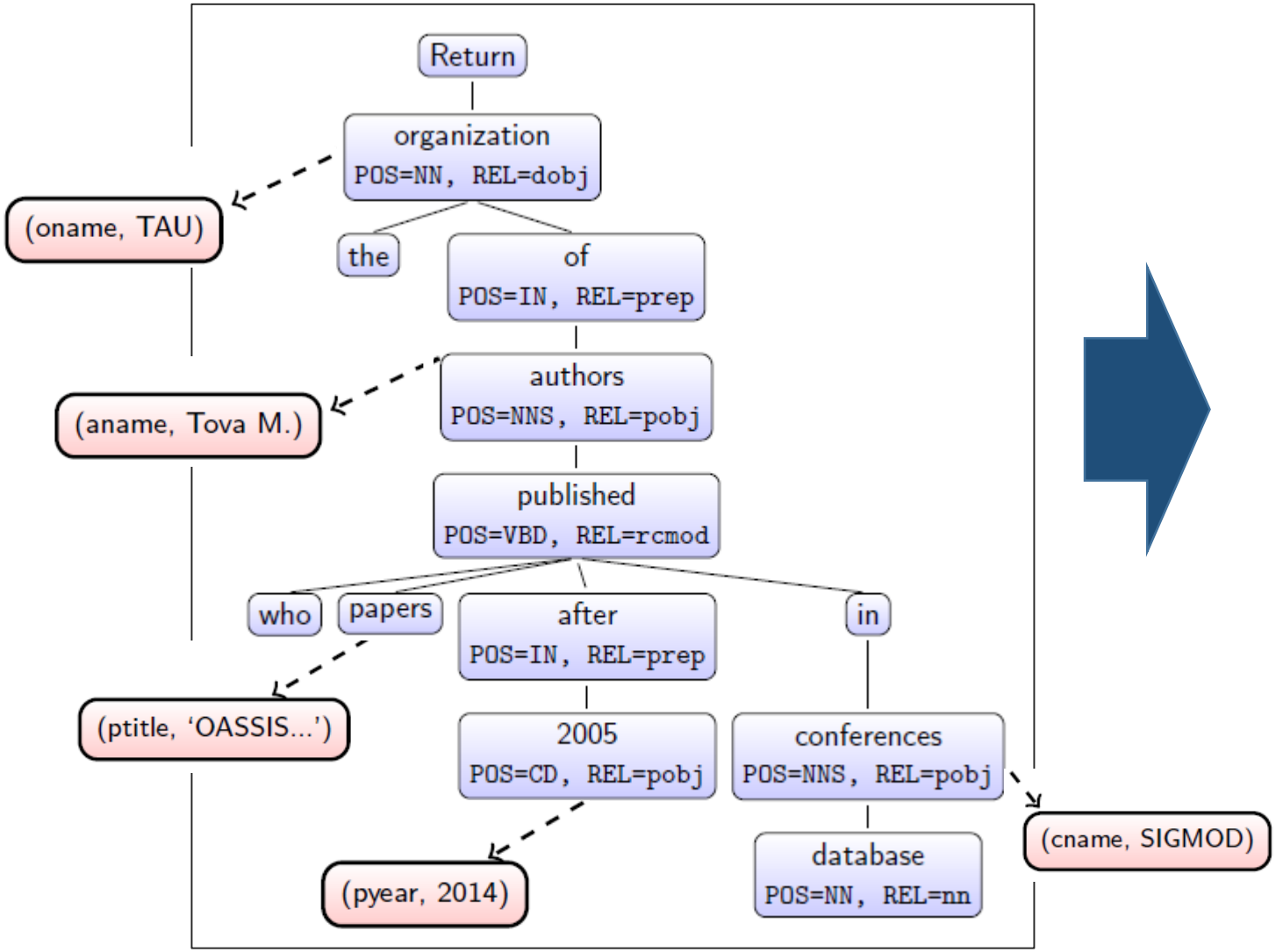
Goal:

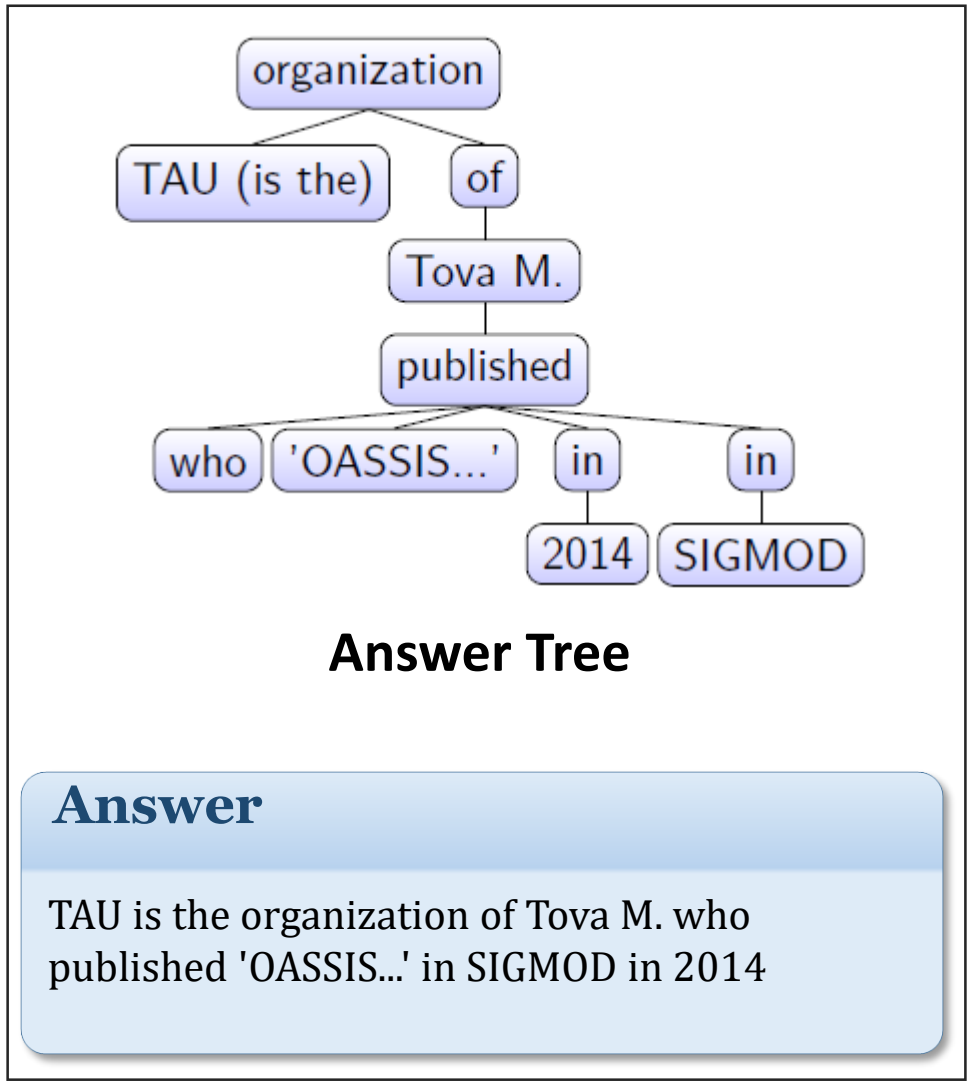
- Generate Answer Tree with explanation

Algorithm:

- Start with root node.
- **Case 1: Leaf node –** replace the value mapped from dependency to query mapping and assignment
- **Case 2: Modifier -** replace entire subtree with value mapped to it.
- **Case 3: Obj has non-verb mod child -** remove return node and add mapped value as the child
- **Case 4: Obj has verb mod child -** replace the node with its value
- Recursion over all the child nodes.

Mapping of Answer Tree - Example





Answer Tree

Answer

TAU is the organization of Tova M. who published 'OASSIS...' in SIGMOD in 2014

But there is a problem!!

What if there are multiple assignments??

General Case - Multiple Assignments

Provenance consisting multiple assignments:

Provenance

(oname,TAU).(aname,Tova M.).(ptitle,OASSIS...).(cname,SIGMOD).(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,Querying...).(cname,VLDB).(pyear,06')+
(oname,TAU).(aname,Tova M.).(ptitle,Monitoring..).(cname,VLDB).(pyear,07')+
(oname,TAU).(aname,Slava N.).(ptitle,OASSIS...).(cname, SIGMOD)(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,A sample...). (cname,SIGMOD).(pyear,14')+
(oname,UPENN).(aname,Susan D.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')

How to handle this ?

General Case - Provenance Factorization

Idea:

- Use algebraic factorization to take out common terms from provenance

Provenance

(oname,TAU).(aname,Tova M.).(ptitle,OASSIS...).(cname,SIGMOD).(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,Querying...).(cname,VLDB).(pyear,06')+
(oname,TAU).(aname,Tova M.).(ptitle,Monitoring..).(cname,VLDB).(pyear,07')+
(oname,TAU).(aname,Slava N.).(ptitle,OASSIS...).(cname, SIGMOD)(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,A sample...). (cname,SIGMOD).(pyear,14')+
(oname,UPENN).(aname,Susan D.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')

Two Different Factorization

```
[TAU].
      ([Tova M.].
            ([VLDB].
                  ([2006] . [Querying…]
                  +[2007] . [Monitoring…]))
            + [SIGMOD] . [2014] .
                  ([OASSIS…] + [A Simple…]))
      +[Slava N.] . [OASSIS…] . [SIGMOD] . [2014])
+[UPENN] . [Sussen D.] . [OASSIS…] . [SIGMOD]. [2014]
```

```
[TAU] .
      ([SIGMOD] . [2014] .
            ([OASSIS…] .
                  ([Tova M.]
                  + [Slava N.]))
            + [Tova M.] . [A Sample…])
      + [VLDB] . [Tova M.]
            ([2006] . [Querying…].
            + [2007] . [Monitoring…])
+ [UPENN]. [Susan D.]. [OASSIS…]. [SIGMOD]. [2014]
```

Provenance Factorization

How to select factorization?

- It should be short
- Maximum no of appearance of atom is minimal.
- Consistent with NL query (T Compatible)

Factorization Algorithm – (Greedy Approach)

Goal:

- Generate a minimal T-compatible factorization

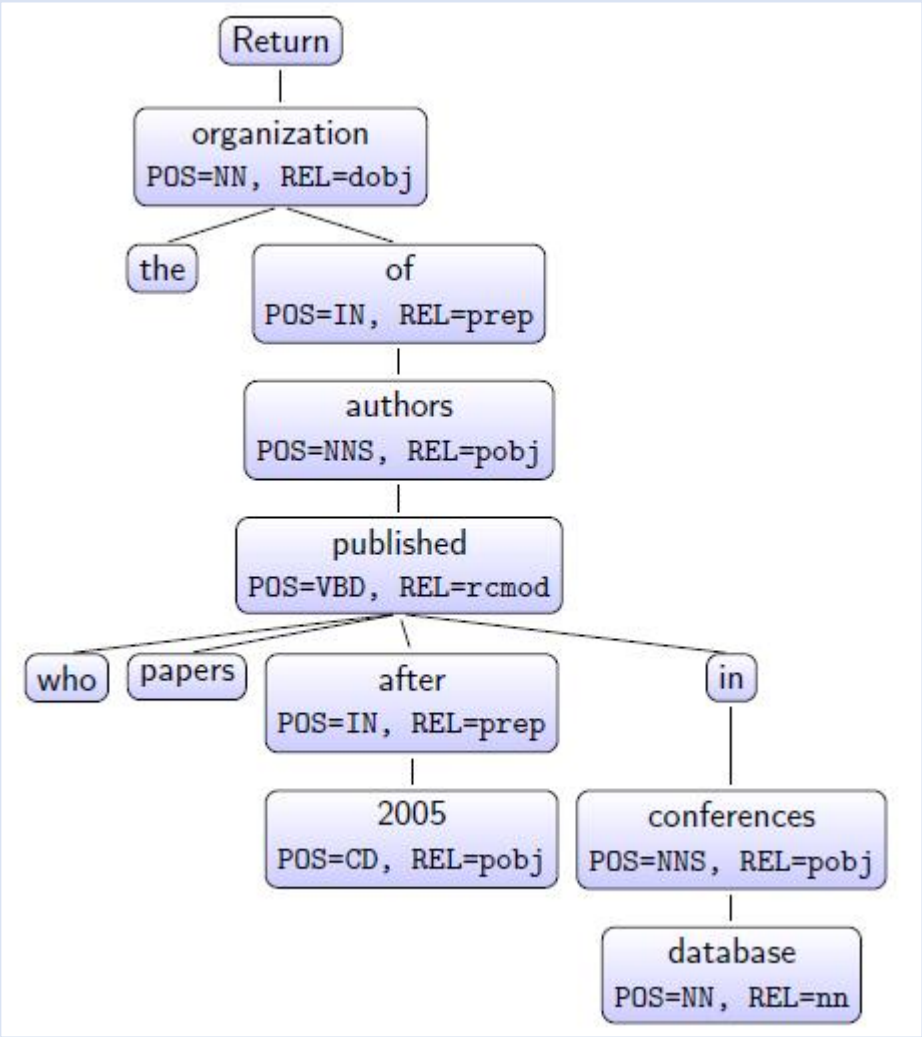
Algorithm:

- Factorize greedily: traverse the dependency tree level-by-level
- For every level with mapped words, factorize their corresponding values in the provenance
- Prioritize which values to take-out in each level by frequency

Time Complexity:

- $O(n^2 \log n)$

Factorization Algorithm - Example



(oname,TAU). (aname,Tova M.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')+
(oname,TAU).(aname,Tova M.). (ptitle,Querying...).(cname,VLDB).(pyear,06')+
(oname,TAU).(aname,Tova M.). (ptitle,Monitoring..).(cname,VLDB).(pyear,07')+
(oname,TAU).(aname,Slava N.).(ptitle,OASSIS...). (cname,SIGMOD)(pyear,14')+
(oname,TAU).(aname,Tova M.).(ptitle,A sample...). (cname,SIGMOD).(pyear,14')+
(oname,UPENN).(aname,Susan D.).(ptitle,OASSIS...). (cname,SIGMOD).(pyear,14')

TAU is the organization of
Tova M. who published
in **VLDB**
'Querying...' in **2006** and
'Monitoring...' in **2007**
and in **SIGMOD** in **2014**
'OASSIS...' and **'A sample...'**
and **Slava N.** who published
'OASSIS...' in **SIGMOD** in **2014**.

UPENN is the organization of **Susan D.** who published **'OASSIS...'** in **SIGMOD** in **2014**.

Why to look for compatibility ?

Shorter Factorization

```
[TAU]
        ([SIGMOD]  [2014]
                  ([OASSIS...]
                            ([Tova M.] + [Slava N.]))
                  + [Tova M.]  [A Sample...])
       + [VLDB]  [Tova M.]
                  ([2006]  [Querying...]
                 + [2007]  [Monitoring...])
  + [UPENN]  [Susan D.]  [OASSIS...]  [SIGMOD]  [2014]
```

Sentence

```
TAU is the organization of authors who published in
     SIGMOD 2014
          'OASSIS...' which was published by
                Tova M. and Slava N.
           and Tova M. published 'A sample...'
     and Tova M. published in VLDB
          'Querying...' in 2014
          and 'Monitoring...' in 2007.
UPENN is the organization of Susan D. who published 'OASSIS...' in SIGMOD in 2014
```

Summarization

Factorization representation may be convoluted:

- When there are more assignments.
- Assignment involves multiple distinct value.

Summarization

Idea:

- Understand which parts of provenance can be grouped together
- Group descendent of node according to their type
- Summarize each group separately

Summarization

Summarized Factorization

(A) [TAU] Size([Tova M.],[Slava N.]) Size([VLDB],[SIGMOD])
Size([Querying...],[Monitoring...],
[OASSIS...],[A Sample...]) Range([2006],[2007],[2014])
(B) [TAU](
[Tova M.]
Size([VLDB],[SIGMOD])
Size([Querying...],[Monitoring...],
[OASSIS...],[A Sample...]) Range([2006],[2007],[2014])
[Slava N.] [OASSIS...] [SIGMOD] [2014])

Summarized Sentences

(A) TAU is the organization of 2 authors who published 4 papers in 2 conferences in 2006 - 2014.

(B) TAU is the organization of Tova M. who published 4 papers in 2 conferences in 2006 - 2014 and Slava N. who published 'OASSIS...' in SIGMOD in 2014.

Examples

Query	Single Assignment	Multiple Assignments - Summarized
Return the homepage of SIGMOD	http://www.sigmod2011.org/ is the homepage of SIGMOD	
Return the authors who published papers in SIGMOD before 2015 and after 2005	Tova M. published "Auto-completion..." in SIGMOD in 2012	Tova M. published 10 papers in SIGMOD in 2006- 2014
Return the authors from TAU who published papers in VLDB	Tova M. from TAU published "XML Repository..." in VLDB	Tova M. from TAU published 11 papers in VLDB
Return the authors who published papers in database conferences	Tova M. "published Auto-completion..." in SIGMOD	Tova M. published 96 papers in 18 conferences
Return the organization of authors who published papers in database conferences after 2005	TAU is the organization of Tova M. who published `OASSIS...' in SIGMOD in 2014	TAU is the organization of 43 authors who published 170 papers in 31 conferences in 2006 - 2015

Conclusion

- First paper on provenance for NL queries
- Using factorization and summarization effectively to make provenance reasonable
- Devised criteria for provenance factorization that accounts for its presentation in NL

Limitations & Future Scope

- Part of design dependent on NaLIR; need to generalize the process.
- Solution is limited to CQ.

Thank You!

Questions?