

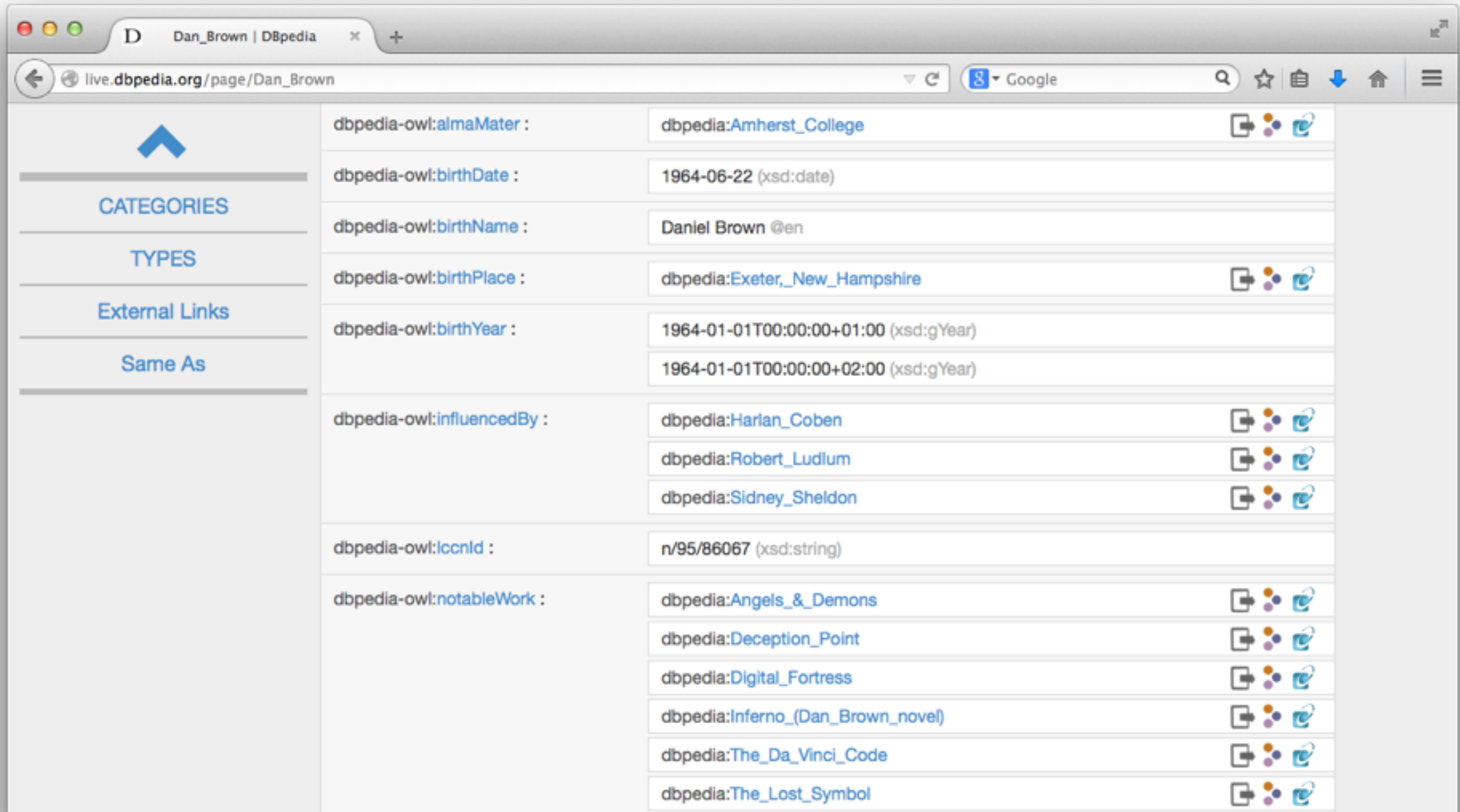
QA over Linked Data using Stanford Dependencies

a project at TU Darmstadt's LT lab by Johannes Simon

The Question Answering Systems Project

- **Goal:** develop an IBM Watson-like QA system
- Individual student projects

Linked Data



The screenshot shows a web browser window with the address bar displaying 'live.dbpedia.org/page/Dan_Brown'. The page title is 'Dan_Brown | DBpedia'. The left sidebar contains navigation links: 'CATEGORIES', 'TYPES', 'External Links', and 'Same As'. The main content area displays a table of properties and values for Dan Brown, with each value being a linked resource.

dbpedia-owl:almaMater :	dbpedia:Amherst_College
dbpedia-owl:birthDate :	1964-06-22 (xsd:date)
dbpedia-owl:birthName :	Daniel Brown @en
dbpedia-owl:birthPlace :	dbpedia:Exeter,_New_Hampshire
dbpedia-owl:birthYear :	1964-01-01T00:00:00+01:00 (xsd:gYear) 1964-01-01T00:00:00+02:00 (xsd:gYear)
dbpedia-owl:influencedBy :	dbpedia:Harlan_Coben dbpedia:Robert_Ludlum dbpedia:Sidney_Sheldon
dbpedia-owl:isccnId :	n/95/86067 (xsd:string)
dbpedia-owl:notableWork :	dbpedia:Angels_&_amp;_Demons dbpedia:Deception_Point dbpedia:Digital_Fortress dbpedia:Inferno_(Dan_Brown_novel) dbpedia:The_Da_Vinci_Code dbpedia:The_Lost_Symbol

Linked Data

Green Day - Relationships... x

musicbrainz.org/artist/084308bd-1654-436f-ba03-df6697104e19/relationships

Google

MusicBrainz

search Artist Search

About Blog Products Search Documentation Contact Us

Log In Create Account

Green Day
~ Group

Overview Releases Recordings Works **Relationships** Aliases Tags Details Edit

Relationships

founding members: [John Kiffmeyer](#) (1987 – 1990)
[Billie Joe Armstrong](#) (1987 –)
[Mike Dirnt](#) (1987 –)

members: [Tré Cool](#) (1990 –)
[Jason White](#) (US guitarist for Green Day & Pinhead Gunpowder) (2012 –)

supporting artists: [Garth Schultz](#) (1997 – 1999)
[Tim Chunks](#) (1997 – 1999)
[Gabrial McNair](#) (1999 – 2001)
[Kurt Lohmiller](#) (1999 – 2004)
[Jason White](#) (US guitarist for Green Day & Pinhead Gunpowder) (1999 – 2012)
[Jeanne Geiger](#) (2001)
[Jason Freese](#) (2003 –)
[Mike Pelino](#) (2004 – 2005)
[Ronnie Blake](#) (2004 – 2005)
[Jeff Matika](#) (2009 –)

signed by: [Lookout! Records](#) (1989 – 1993)
[Reprise Records](#) (1993 –)

Allmusic: <http://www.allmusic.com/artist/mn0000154544> [info]
BBC Music: <http://www.bbc.co.uk/music/artists/084308bd-1654-436f-ba03-df6697104e19> [info]
Discogs: <http://www.discogs.com/artist/251593> [info]




Image from Wikimedia Commons

Artist information

Type: Group

Founded: 1989 (25 years ago)

Founded in: [Berkeley, California, United States](#)

Area: [United States](#)

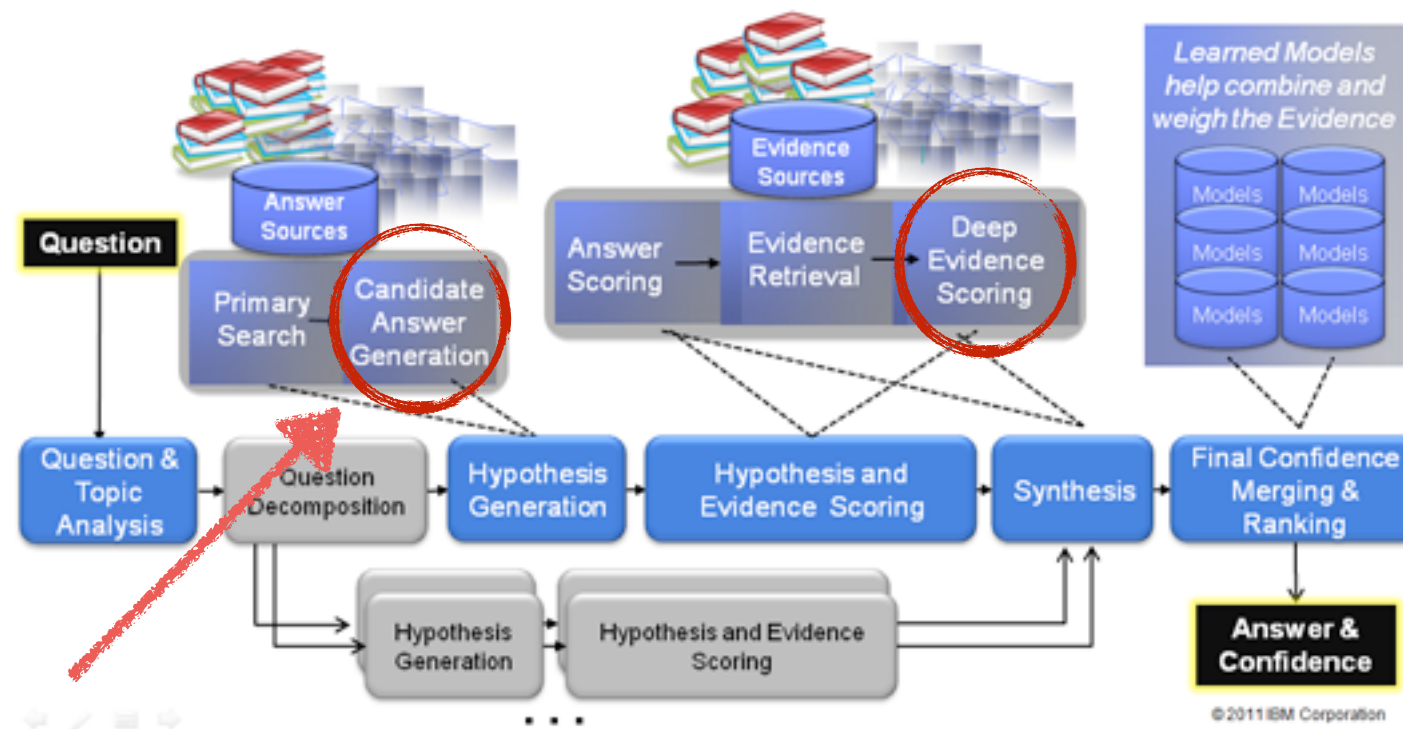
ISNI code: 0000 0001 2271 1282

Rating
★★★★★ (see all ratings)

Tags
[pop punk](#), [american](#), [punk rock](#), [rock](#), [pop](#), more...

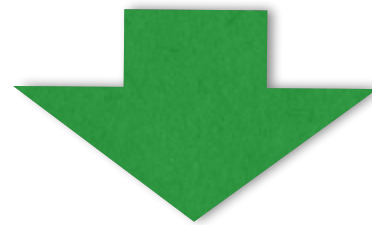
External links

IBM Watson



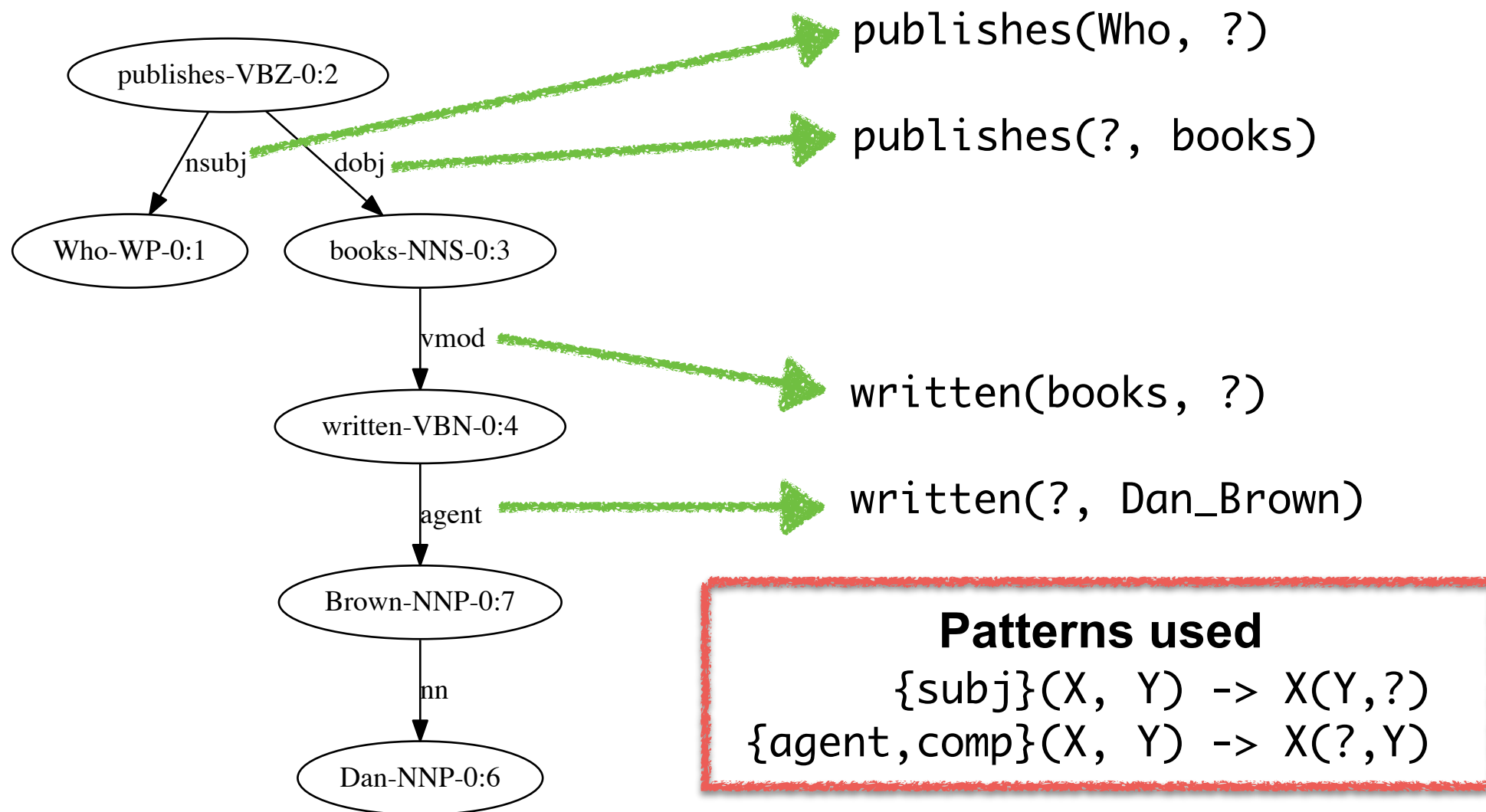
QA over Linked Data

"Who publishes books written by Dan Brown?"



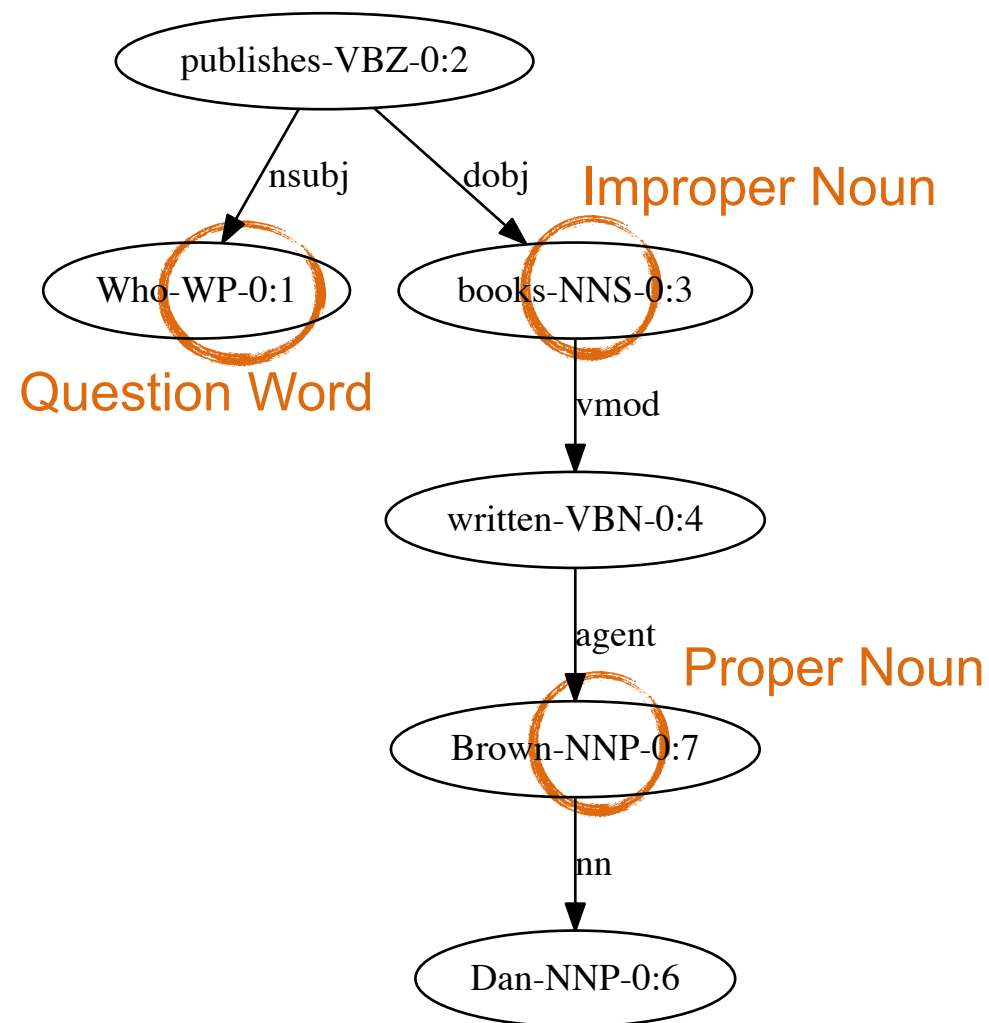
```
SELECT ?who WHERE {  
  ?who dbo:publisher ?book .  
  ?book dbo:author dbpedia:Dan_Brown .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

The Approach (1)



"Who publishes books written by Dan Brown?"

The Approach (2)



publishes(Who, ?)
publishes(?, books)
written(books, ?)
written(?, Dan_Brown)

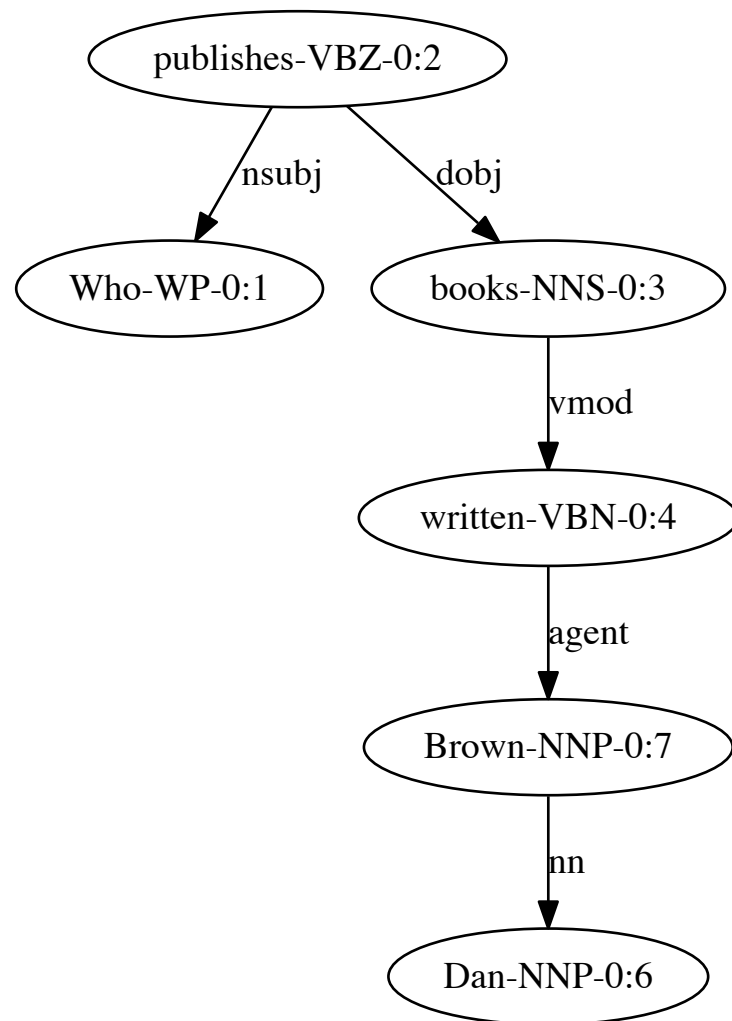


publishes(Who, books)
written(books, Dan_Brown)



```
SELECT ? WHERE {  
  ?who "publish" ?book .  
  ?book "write" "Dan Brown" .  
  ?who a ? .  
  ?book a ? .  
}
```


The Approach (3)



Question Focus

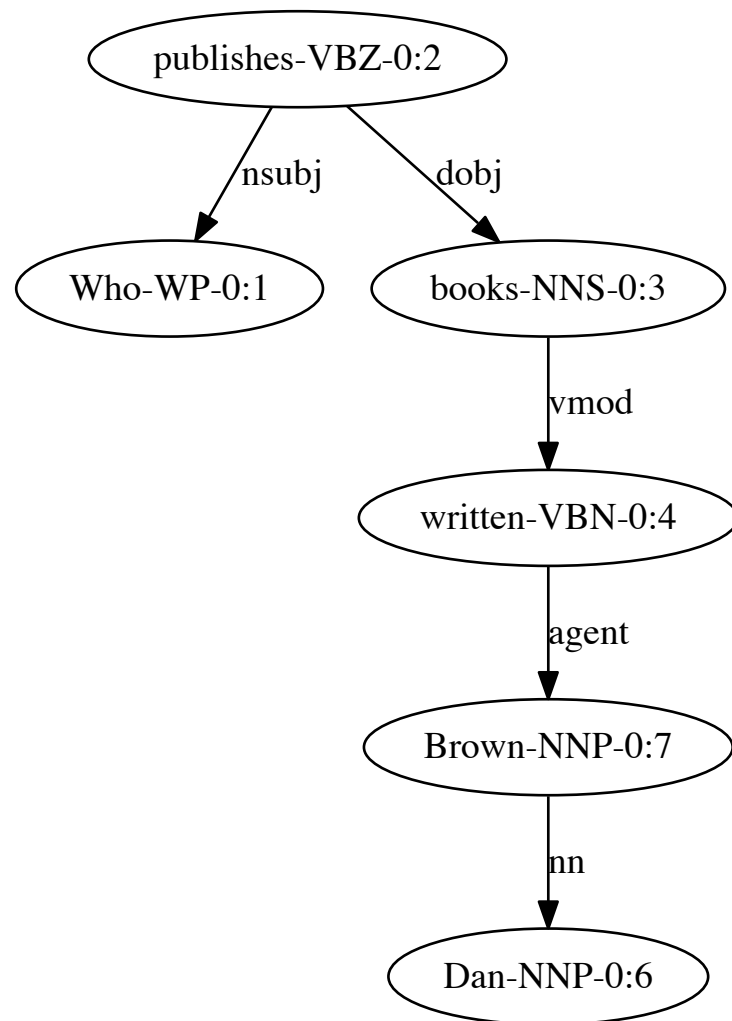
```
SELECT ? WHERE {  
  ?who "publish" ?book .  
  ?book "write" "Dan Brown" .  
  ?who a ? .  
  ?book a ? .  
}
```

Variable Types



```
SELECT ?who WHERE {  
  ?who "publish" ?book .  
  ?book "write" "Dan Brown" .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

The Approach (4)



```
SELECT ?who WHERE {  
  ?who "publish" ?book .  
  ?book "write" "Dan Brown" .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

```
SELECT ?who WHERE {  
  ?who dbo:publisher ?book .  
  ?book dbo:author dbpedia:Dan_Brown .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

Candidate Query Generation

```
SELECT ?who WHERE {  
  ?who dbo:publisher ?book .  
  ?book dbo:author dbpedia:Dan_Brown .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

 (0.262)

```
SELECT ?who WHERE {  
  ?who dbo:publisher ?book .  
  ?book dbo:writer dbpedia:Dan_Brown .  
  ?who a schema:Person .  
  ?book a bibo:Book .  
}
```

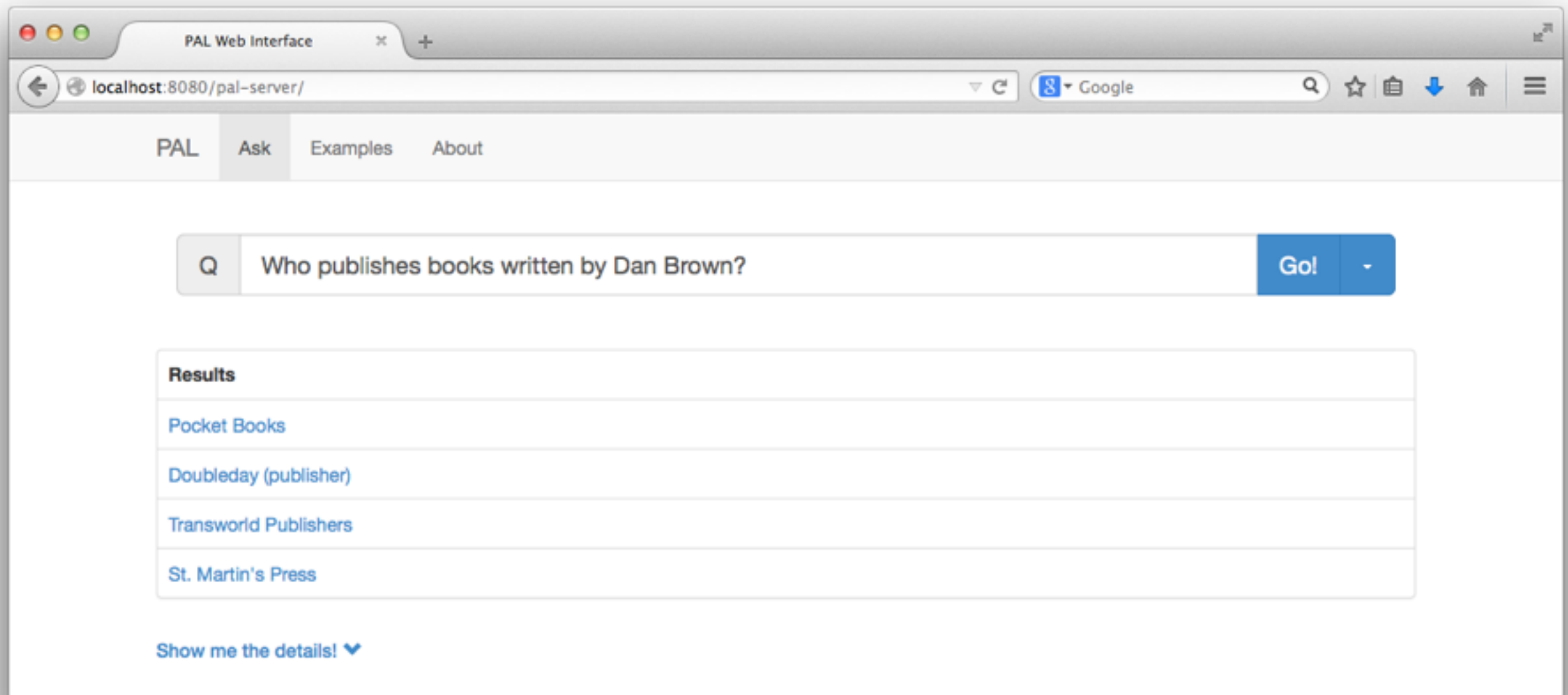
 (0.262)

```
SELECT ?who WHERE {  
  ?who dbo:publisher ?book .  
  ?book dbo:writer dbpedia:Dan_Brown .  
  ?who a schema:Organization .  
  ?book a bibo:Book .  
}
```

 (0.131)

...

What does it look like?



What does it look like?

The screenshot displays the PAL Web Interface in a web browser. The browser's address bar shows the URL `localhost:8080/pal-server/`. The interface has a navigation bar with links for **PAL**, **Ask**, **Examples**, and **About**. The main content area is titled **Input Interpretation** and contains two sections of input analysis.

Input Interpretation

The first section shows the input `book` and `publish`. The `publish` input is interpreted as `publisher (related form)`, which is circled in orange, and then mapped to the URI `dbpedia-owl:publisher (URI match)`. The input `who` is also shown.

The second section shows the input `book` and `write`. The `write` input is interpreted as `author (Hyponym)`, which is circled in orange, and then mapped to the URI `dbpedia-owl:author (URI match)`. The input `Dan Brown` is mapped to `dbpedia:Dan_Brown (partial match)`.

Variable Type Constraints

The third section shows the variable type constraints for the inputs. The input `book` is constrained to `bibo:Book (URI match)`. The input `who` is constrained to `schema:Organization ("who")`.

Evaluation

- QALD-2 challenge
- 200 questions for DBpedia, 100 for training/test each
- also 200 questions for MusicBrainz, but queries were more complex and not tested against

Evaluation Results

Test set	# Correct	Partially correct	Precision	Recall	F-1 score
Training	27	9	76.8	32.2	45.3
Test	16	5	65.1	20.0	30.6

System	# Correct	Partially correct	Precision	Recall	F-1 score
SemSeK	32	7	44.0	48.0	46.0
Alexandria	5	10	43.0	(46.0)	(45.0)
MHE	30	12	36.0	40.0	38.0
QAKis	11	4	39.0	(37.0)	(38.0)
Hakimov	15	?	>83.3	>15.0	>25.4

$$\text{Recall} = \frac{\text{number of correct system answers}}{\text{number of gold standard answers}}$$

$$\text{Precision} = \frac{\text{number of correct system answers}}{\text{number of system answers}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

*"... computes the overall **precision** and **recall** taking the average **mean of all single precision and recall values**, as well as the overall F-measure."*

48% recall from < 39% correctly answered questions?

Problematic Questions

Which state of the USA has the highest population density?



```
SELECT ?state WHERE {  
  ?state "highest population density" ?x  
  ?state "state" dbpedia:United_States  
  ?state a dbpedia-owl:Country .  
}
```

```
SELECT ?state WHERE {  
  ?state a yago:StatesOfTheUnitedStates .  
  ?state dbp:densityrank ?rank .  
}  
ORDER BY ASC(?rank) LIMIT 1
```


Outlook

- Future work:
 - Match entire phrases to entities and classes (e.g. YAGO classes)
 - Add mechanism to handle comparison queries
 - Match literals (e.g. dates), not only URIs
 - more suggestions, see report

Get it on github

- <https://github.com/johannessimon/pal>
 - most code
 - project report
- <https://github.com/johannessimon/pal-server>
 - web frontend