Rebele

The YAGO knowledge base

Dutlin

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Uni Shell

onclusion



Expanding the YAGO knowledge base



Télécom ParisTech

2018-07-19

What is a knowledge

What is YAGO?

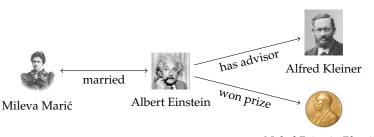
Outline

Using YAGO for the humanities

Adding Words to

Answering Queries with Unix Shell

Conclusion



Nobel Prize in Physics

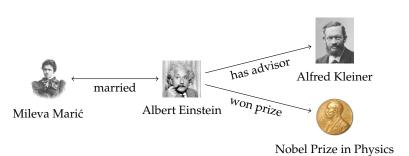
What is a knowledge base?

Expanding the YAGO knowledge base

Rehele

What is a knowledge





Applications of knowledge bases:

- Question answering
- Semantic search
- Text analysis
- Machine translation

Adding Words to

Answering Queries with Unix Shell

Conclusion

Knowledge base with 10 million entities and >210 million facts

- Automatically extracted from Wikipedia, Wordnet, and Geonames
- ► Multilingual facts from 10 languages
- Focus on precision
- Developed by Max-Planck Institute for Informatics and Télécom ParisTech























Languages

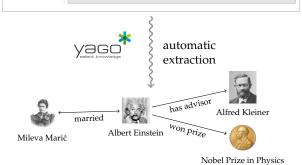
English German

French



What is YAGO?





Categories: Nobel laureates in Physics

Albert Einstein

Doctoral advisor(s) Alfred Kleiner

Mileva Marić

Spouse(s)

Albert Einstein

Albert Einstein was a

physicist. His work influenced the philosophy of science. He developed

the theory of relativity.

Involvement

Expanding the YAGO knowledge base

Rebele

knowledge base What is a knowledge

What is YAGO Involvement

Julinic

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Uni

Conclusion

I joined the project in 2014

Maintenance and development

Contributed to open source release in 2017 at https://github.com/yago-naga/yago3/

Coordinated / contributed to the evaluation

ground truth: Wikipedia

▶ 98% facts of the sample were correct

Publication: ISWC 2016 (resource paper)



Fabian Suchanek









Gerhard We

YAGO is very accurate. But how complete is it?

legexes

Answering Queries with Uni: Shell

Conclusion

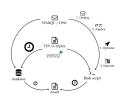
Contributions:

Extracting more information about residences, gender, birth and death dates

Repairing regular expressions by adding missing words

1 3 8 0 9 78 9 79 d d ... d d

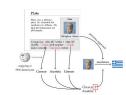
Preprocessing tabular data by transforming queries to Bash scripts



Adding Words to Legexes

Answering Queries with Uniz Shell

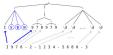
Conclusion



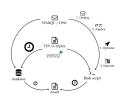
Repairing regular expressions by adding missing words

Extracting more information about residences, gender, birth and death dates

Contributions:



Preprocessing tabular data by transforming queries to Bash scripts

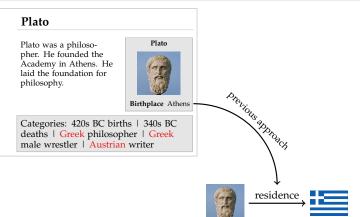


Problem statement

Every person lives somewhere, but YAGO knows the residence only for 30% of the people

Every person has a gender, but YAGO knows the gender only for 64% of the people

How can we make YAGO more complete?



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Extensions

Place of residence

Evaluation

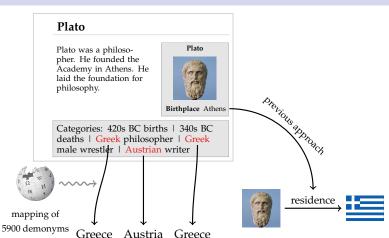
Births per month

Relative population size

Summa

Adding Words to Regexes

Answering Queries with Unix Shell



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Extensions
Place of residence

Gender

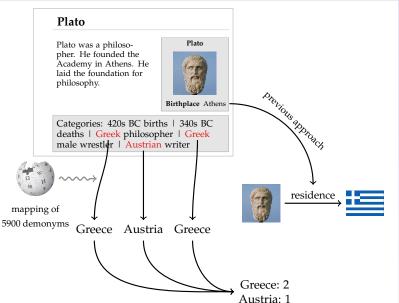
CVAIUAUOII

Life span over time Relative population

Summary

Adding Words to

Answering Queries with Unix



Expanding the YAGO knowledge base

Rebele

The YAGO

Outlin

Using YAGO for the humanities

Place of residence

Gender

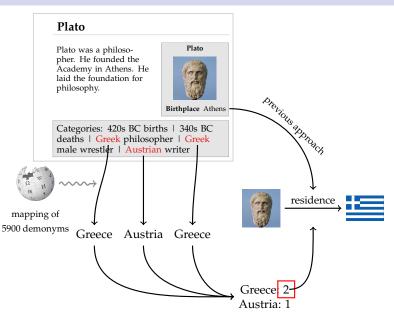
Evaluation p: d

Life span over time

Summary

Adding Words to

Answering Queries with Unix



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Extensions

Place of residence

riace of residence Gender

Evaluation

Life span over time

Summary

Adding Words to Regexes

Answering Queries with Unix Shell

Using YAGO for the humanities: Gender

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for

Extensions

Place of residence

Gender

Evaluation

Births per month

Relative population size

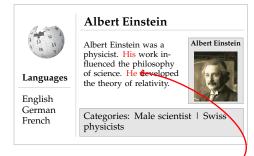
Summa

Adding Words to

Answering Queries with Unit

Conclusion

Extract gender:



From pronoun:

- YAGO's original algorithm
- Count pronouns (he, him / she, her)
- Assign gender accordingly

9/41

Using YAGO for the humanities: Gender

Expanding the YAGO knowledge base

Rebele

he YAGO nowledge base

Outlin

Using YAGO for the humanities

Extensions

Extensions

Place of resideno Gender

Evaluatio

Evaluation

Life span over time

Relative population size

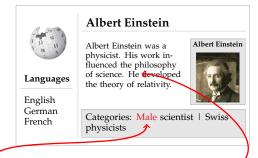
Summa

Adding Words to Regexes

Answering Queries with Unix

Conclusion

Extract gender:



From category

From pronoun:

- YAGO's original algorithm
- Count pronouns (he, him / she, her)
- Assign gender accordingly

Using YAGO for the humanities: Gender

Expanding the YAGO knowledge base

Rebele

The YAGO mowledge base

Outlin

Using YAGO for

Problem stateme

Place of residence

Gender

Evaluation

Life span over time

Relative population size

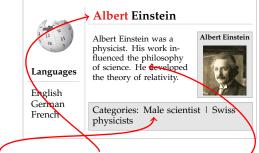
Summa

Adding Words t Regexes

Answering Queries with Unix Shell

Conclusion

Extract gender:



From category

From first name:

- Count males/females for each first name
- Assign names to gender accordingly

From pronoun:

- YAGO's original algorithm
- Count pronouns (he, him / she, her)
- Assign gender accordingly

Using YAGO for the humanities: Evaluation

- Compare extraction process on Wikipedia dump from 2017-02-20
- ► Extracted on 11 languages
- ▶ Evaluate precision based on a sample of 100 people

Extraction	YAGO before	Recall	YAGO now	Recall	Precision	DBpedia (en)
Place of residence	0.7m	30%	2.1m	91% (+201%)	97% (*)	0.7m
Gender	1.5m	64%	2.0m	87% (+35%)	98%	4k
Birth dates	1.6m	69%	1.7m	74% (+8%)	100%	0.8m
Death dates	0.7m	33%	0.8m	36% (+10%)	100%	0.3m

Table: Coverage and precision of our methods.

Recall relative to total number of people in YAGO (2.2m).

m million k thousand 10/41 (*) 6% of anachronistic residencies (e.g., German Empire instead of Germany)

Using YAGO for the humanities: Births per month

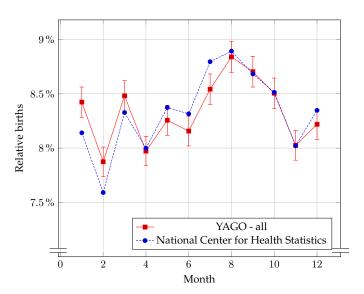


Figure: Births per month in the United States between 2003 and 2015 (with the Student's t confidence interval at $\alpha=95\%$).

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for

Problem statem

Dlace of recid

ender

Births per month

Life span over time Relative population

Summary

Adding Words to

Answering Queries with Unix Shell

Using YAGO for the humanities: Births per month

Expanding the YAGO knowledge base

Rehele

Births per month

Relative age effect

The relative age effect describes a bias. People born early in the selection period of sports or academia are more likely to succeed.

Languages

English Euskara



Categories: Ageism | Epidemiology

7.5% Natior 2

9%

8.5%

8%

Relative births

Figure: Births per month in the United Sta (with the Student's t confidence interval a

Using YAGO for the humanities: Births per month

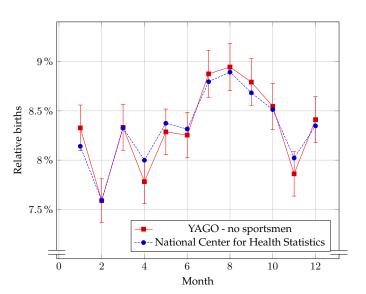


Figure: Births per month in the United States between 2003 and 2015 (with the Student's t confidence interval at $\alpha=95\%$).

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outli

Using YAGO for

r robiem staten

Extensions Place of residenc

raluation

Births per month

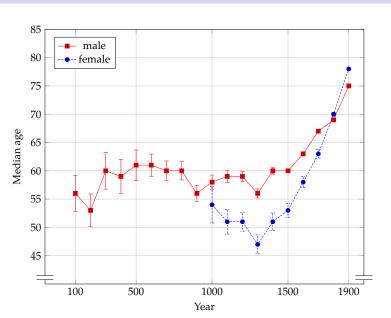
Life span over time Relative population size

Summary

Adding Words to Regexes

Answering Queries with Unix Shell

Using YAGO for the humanities: Life span over time



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outli

Using YAGO for the humanities

Extensions

Evaluation

Births per month

Life span over time Relative population

ummary

Regexes

Answering Queries with Unix Shell

onclusion

Figure: Median age over time, by year of birth

12/41

Using YAGO for the humanities: Relative population size



Rebele

The YAGO knowledge base

Outline

Using YAGO f

Problem statement

Place of residence

Evaluation

Births per month

Relative population

ummary

Adding Words

Answering Queries with Unit

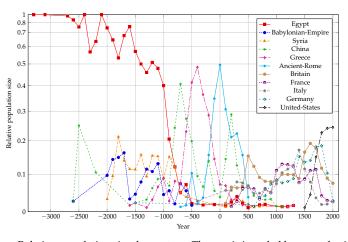
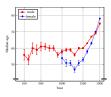


Figure: Relative population size, by century. The *y*-axis is scaled by a quadratic function.

Using YAGO for the humanities: Summary

- Extension of YAGO:
 - ► More people with residences (+201%, 97% precison)
 - ► More people with genders (+35%, 98% precision)
 - ► More birth and death dates (+8%/10%, 100% precision)
- Case studies:
 - ▶ Births per month
 - Life span over time
 - Relative population size over time
- ► Interdisciplinary project



Publication: ISWC 2017 (workshop paper)



Thomas Rebele



Arash Nekoei



Fabian Suchanek

We often had to repair regular expressions (e.g., for matching dates).

Can we automate this step?

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Extensions

Gender

Evaluation

Life span over time Relative population

size Summary

Adding Words to Regexes

Answering Queries with Unit Shell

Using YAGO for the humanities: Summary

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO

Extensions

'lace of reside Gender

valuation

Births per month Life span over time

Relative populatio size

Summary

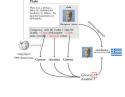
Adding Words to Regexes

Answering Queries with Unix

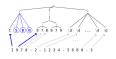
Conclusion

Contributions:

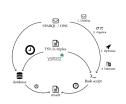
Extracting more information about residences, gender, birth and death dates



Repairing regular expressions by adding missing words



Preprocessing tabular data by transforming queries to Bash scripts



Adding Words to Regexes: Introduction





Why does YAGO not know the ISBN numbers of my books?

- ▶ We want to find ISBN numbers in Wikipedia to include it in YAGO
- ► We try the regex

ISBN(978|979)?\d{10}



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO

Regexes

Introduction

Troblem statement

What is new in our

approach Approximate rege

natching inding the gaps

add missing par eedback functio

Experiments Summary

Answering Queries with Unix Shell

Adding Words to Regexes: Introduction





Why does YAGO not know the ISBN numbers of my books?

- ▶ We want to find ISBN numbers in Wikipedia to include it in YAGO
- ► We try the regex ISBN(978|979)?\d{10}
- ► Why does the regex not find 1978-2-1234-5680-3
- ▶ How can we modify the regex automatically to match the word?



Expanding the YAGO knowledge base

Rehele

The YAGO knowledge base

Outlin

Using YAGO fo

Regexes

Introduction

Vhat is new in our

ipproach

matching

Add missing parts Feedback function

Experiments Summary

Answering Queries with Unix Shell

Problem statement, first try:

Given

- ▶ a regular expression *r* and
- \triangleright a set of positive examples E^+ ,

find a regular expression r' such that

- $ightharpoonup L(r) \subseteq L(r')$
- $ightharpoonup E^+ \subseteq L(r')$

ISBN(978|979)?\d{10} { I978-2-1234-5680-3 }

Adding Words to Regexes: Problem statement

Expanding the YAGO knowledge base

Rehele

ISBN(978|979)?\d{10}

{ 1978-2-1234-5680-3 }

Problem statement

Problem statement, first try:

Given

- a regular expression r and
- \triangleright a set of positive examples E^+ ,

find a regular expression r' such that

- $ightharpoonup L(r) \subset L(r')$
- $ightharpoonup E^+ \subset L(r')$

Solution:

Given

- \triangleright a regular expression r,
- \triangleright a set of positive examples E^+ ,
- \triangleright a set of negative examples E^- ,

find a regular expression r' such that

- $ightharpoonup L(r) \subset L(r')$
- $ightharpoonup E^+ \subset L(r')$
- $ightharpoonup L(r') \cap E^-$ is small

ISBN(978|979)?\d{10} { 1978-2-1234-5680-3 } { 0612345678 }

Problem statement

Given

- a regular expression r,
- \triangleright a set of positive examples E^+ ,
- \triangleright a set of negative examples E^- ,

find a regular expression r' such that

- $ightharpoonup L(r) \subset L(r')$
- $ightharpoonup E^+ \subset L(r')$
- ► $L(r') \cap E^-$ is small

ISBN(978|979)?\d{10}

{ 1978-2-1234-5680-3 }

{ 0612345678 }

Answering Queries with Uni Shell

Evaluation:

- ▶ Precision of r' ≥ or ≈ precision of r
- Recall of $r' \ge \text{recall of } r$ (w.r.t. the intended meaning of the regex)

onclusion

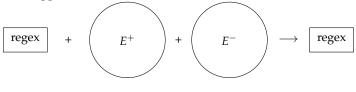
Problem statement

18/41

What is new in our

approach

Previous approaches:



Our approach:

$$\boxed{\text{regex}} \quad + \qquad \boxed{E^+} \qquad + \qquad \boxed{E^-} \qquad \rightarrow \qquad \boxed{\text{regex}}$$

Rationale: creating a large set of positive examples is difficult

Step 1: match string and regex approximately [Myers et al. 1989]

Outlin

Using YAGO for

Adding Words to Regexes

Introduction

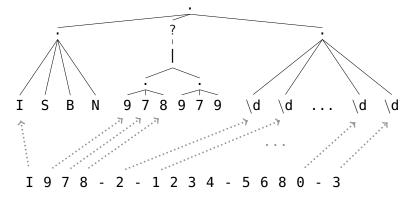
What is new in our approach

Approximate regex matching

Finding the gaps
Add missing parts
Feedback function
Experiments

Answering Queries with Unix

. . .



Outlin

Using YAGO for the humanities

Adding Words to Regexes

Introduction
Problem statemen

What is new in our approach

matching

Finding the gaps

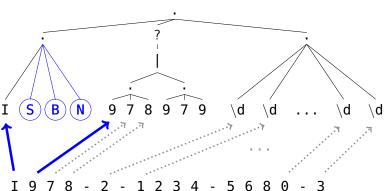
Feedback function Experiments

Answering Queries with Unix

Complusion



▶ Between regex leaves



Using YAGO for the humanities

Adding Words to Regexes

Introduction
Problem statemen

What is new in o approach

matching Finding the gaps

Finding the gaps

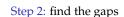
Add missing parts

Feedback function Experiments

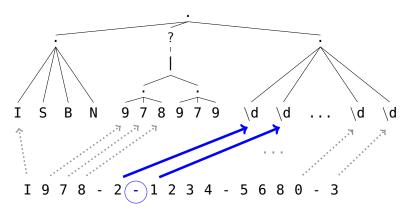
Answering Queries with Univ Shell

Complusion

21/41



- Between regex leaves
- ▶ Between characters of the string



Adding Words to Regexes

Introduction
Problem statement

What is new in our approach

Approximate reg

Add missing parts

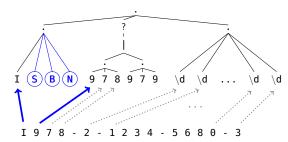
Experiments

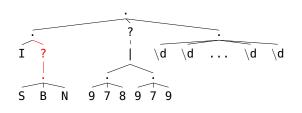
Answering Queries with Unit

Complession

22/41

Step 3 (simple approach): adapt regex, so that it includes the missing parts





Using YAGO for the humanities

Adding Words to Regexes

Introduction
Problem statemen

what is new in our approach Approximate reger

natching Finding the gaps

Add missing parts Feedback function

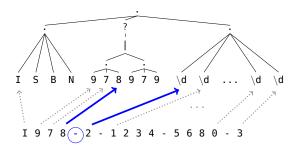
Experiments
Summary

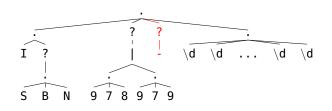
Answering Queries with Unix Shell

Complession

22/41

Step 3 (simple approach): adapt regex, so that it includes the missing parts





hutlina

Using YAGO for

Adding Words to Regexes

Introduction Problem statement

ipproach Approximate regex

matching Finding the gaps

Add missing parts Feedback function Experiments

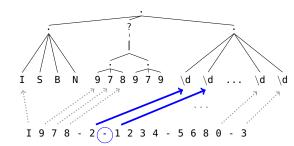
Summary

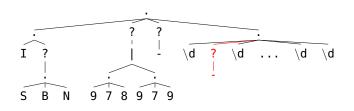
Shell

Conclusion

22/41

Step 3 (simple approach): adapt regex, so that it includes the missing parts





Jutline

Using YAGO for the humanities

Adding Words to Regexes

Introduction Problem statement

approach Approximate rege

matching
Finding the gaps

Add missing parts Feedback function

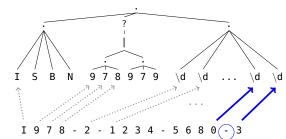
Experiments

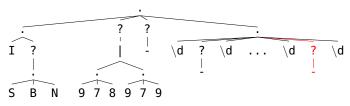
Answering Queries with Unix Shell

Conclusion



Step 3 (simple approach): adapt regex, so that it includes the missing parts





Adding Words to Regexes

Problem statement

approach
Approximate rege

Finding the gaps

Add missing parts Feedback function

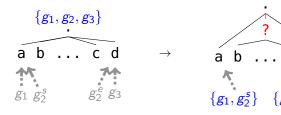
Experiments Summary

Answering Queries with Unit Shell

Conclusion

Step 3 (adaptive approach): adapt regex, so that it includes the missing parts

Exemplarily for a concatenation:



Adding Words to Regexes

Problem statement

approach
Approximate rege

Finding the gaps

Add missing parts Feedback function

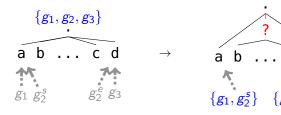
Experiments Summary

Answering Queries with Unit Shell

Conclusion

Step 3 (adaptive approach): adapt regex, so that it includes the missing parts

Exemplarily for a concatenation:



Feedback function

Now we want to find URLs:

- ▶ We try regex $r = http://[a-zA-Z\.]+$
- ▶ It does not find s = wikipedia.org
- Repaired regex $r' = (http://)?[a-zA-Z\.]+$

Problem:

- ightharpoonup r' finds all words
- Precision drops

Adding Words to Regexes: Feedback function

Expanding the YAGO knowledge base

Rehele

Now we want to find URLs:

- ▶ We try regex $r = http://[a-zA-Z\.]+$
- ightharpoonup It does not find s = wikipedia.org
- Repaired regex r' = (http://)?[a-zA-Z].]+

Problem:

- r' finds all words
- Precision drops

Solution: use feedback on set of negative examples E⁻

- Determine the parts of the regex that we can make optional
- We use the number of false positives, i.e.,

$$f(r') = |E^- \cap L(r')| \le \alpha |E^- \cap L(r)|$$

▶ If f(r') = false, add the word as disjunction instead: http://[a-zA-Z\.]+|wikipedia.org

Using YAGO for

Adding Words to Regexes

Introduction
Problem statemen

What is new in our approach

matching Finding the gaps

Add missing part Feedback function

Experiments

ummary

Answering Queries with Unix Shell

Conclusion

Input data:

Datasets:

ReLIE [Li et al., 2008], Enron [Babbar et al., 2010], and Wikipedia infobox attributes

▶ In total 8 tasks (e.g., phone numbers, software names, dates)

► In total 52 regexes

Approaches:

- ightharpoonup Dis: $r|s_1|\cdots|s_n$
- ▶ Star: .*
- ▶ B&S: [Babbar et al., 2010] (reimplementation)
- ▶ Simple
- Adaptive

mput data.	
Datasets:	
Dollie II : ot ol	20001

- ReLIE [Li et al., 2008],
 Enron [Babbar et al., 2010], and
 Wikipedia infobox attributes
- ► In total 8 tasks (e.g., phone numbers, software names, dates)
- ▶ In total 52 regexes

Input data

	baseline				adaptive			
measure	original	dis	star	B&S	simple	$\alpha = 1.0$	$\alpha = 1.1$	$\alpha = 1.20$
F1	55	55	21	40	56	<u>60</u>	<u>60</u>	<u>60</u>
recall	66	67	62	35	69	75	76	<u>77</u>
precision	64	64	14	<u>71</u>	64	63	63	63
length	56	270	2	3929	250	<u>76</u>	80	81

Table: Averaged measures for the different systems. Length is # of characters of the regex.

Adding Words to Regexes: Summary

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Dutline

Using YAGO for the humanities

Adding Words to Regexes

Introduction Problem statem

pproach

atching nding the gaps

Add missing pa

Experiments

Summary

Answering Queries with Unix Shell

Conclusion

Summary:

- Algorithm for adding missing words to regexes
- Increases recall, while keeping precision stable
- Source code available at https://github.com/thomasrebele/regex-repair

Future work:

- ▶ Decrease dependency on E[−]
- Add a generalization step as postprocessing

Publications: ISWC 2017 (demo), PAKDD 2018 (full paper)



Thomas Rebele



Katerina Tzompanaki



Fabian Suchanek

Now that we have all this data, how can we process it efficiently?

Adding Words to Regexes: Summary

Expanding the YAGO knowledge base

Summary

Rehele

Contributions:

Extracting more information about residences, gender, birth and death dates

Repairing regular expressions by adding missing words

Vacc

Preprocessing tabular data by transforming queries to Bash scripts

28/41

Answering Queries with Unix Shell: Motivation

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Adding Words to Regexes

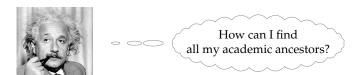
Answering Queries with Uni:

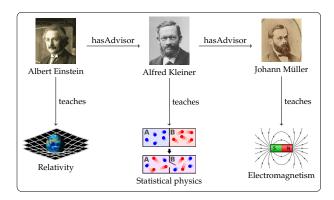
Motivation

Approach

Optimization Experiments

. .





Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for

Adding Words to Regexes

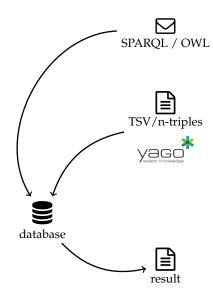
Answering
Queries with Unix

Motiva

Approac

Optimization

Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for

Adding Words to

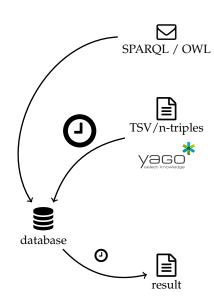
Answering Queries with Unix Shell

Motiva

Approach

Optimization

Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outl

Using YAGO for the humanities

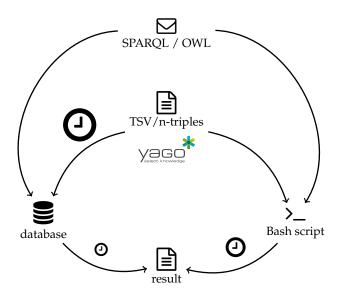
Adding Words to Regexes

Answering
Queries with Unix
Shell

Idea

Approach

Experiments



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outli

Using YAGO for the humanities

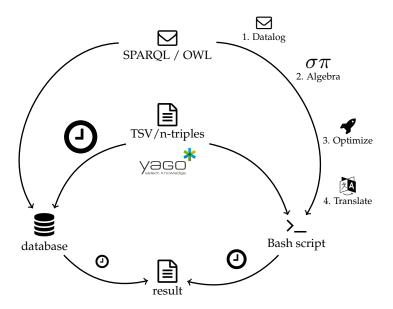
Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach

Experiments



Approach

 $\sigma\pi$ SPAROL / OWI







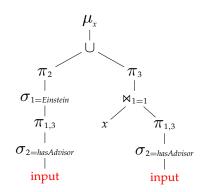
Query "Who are Einstein's academic ancestors?" in SPARQL:

```
SELECT ?Y WHERE {
    <Finstein> <hasAdvisor>+ ?Y
```

Translating the query to Datalog (simplified):

```
facts(X, Y, Z) :~ read ntriples input
adv(X, Y) :- facts(X, "hasAdvisor", Y).
result(Y) :- adv("Einstein", Y).
result(Y) :- result(X), adv(X, Y).
```

```
facts(X, Y, Z) :~ read_ntriples input
adv(X, Y) :- facts(X, "hasAdvisor", Y).
result(Y) :- adv("Einstein", Y).
result(Y) :- result(X), adv(X, Y).
```



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Dutline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

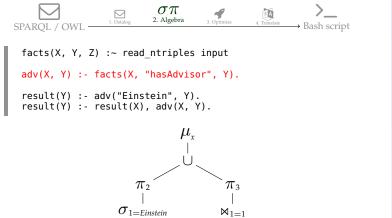
Approach

Optimization Experiments

 $\pi_{1,3}$

 $\sigma_{2=hasAdvisor}$

input



x

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

utline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Experiments Summary

Conclusion

 $\pi_{1,3} \leftarrow ---$ projection

 $\sigma_{2=hasAdvisor} \leftarrow ---$ selection

input

 $\pi_{1,3}$

 $\sigma_{2=hasAdvisor}$

input

 $\sigma\pi$ 2. Algebra 3. Optimize 4. Translate SPAROL / OWL → Bash script facts(X, Y, Z) :~ read ntriples input adv(X, Y) :- facts(X, "hasAdvisor", Y).result(Y) :- adv("Einstein", Y). result(Y) :- result(X), adv(X, Y). π_2 π_3 $\sigma_{1=Einstein}$ $\bowtie_{1=1}$

x

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

utline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach
Optimization
Experiments

Summary

Conclusion

 $\pi_{1,3} \leftarrow ---$ projection

 $\sigma_{2=hasAdvisor} \leftarrow ---$ selection

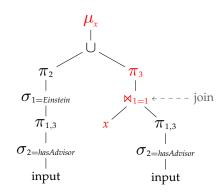
input

 $\sigma\pi$ 2. Algebra SPARQL / OWL

3. Optimize

→ Bash script

facts(X, Y, Z) :~ read ntriples input adv(X, Y) :- facts(X, "hasAdvisor", Y). result(Y) :- adv("Einstein", Y). result(Y) :- result(X), adv(X, Y).



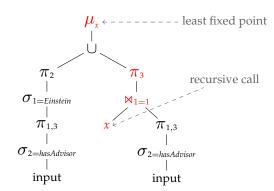
Expanding the YAGO knowledge base

Rebele

Approach



```
facts(X, Y, Z) :~ read_ntriples input
adv(X, Y) :- facts(X, "hasAdvisor", Y).
result(Y) :- adv("Einstein", Y).
result(Y) :- result(X), adv(X, Y).
```



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

utline

Using YAGO for the humanities

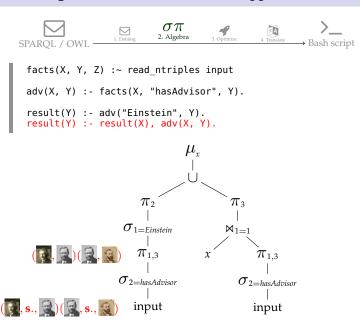
Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Experiments Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Dutline

Using YAGO for the humanities

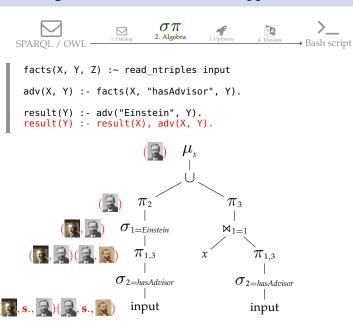
Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization Experiments

Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization Experiments

Summary

 $\sigma\pi$ 2. Algebra 3. Optimize SPAROL / OWL → Bash script facts(X, Y, Z) :~ read ntriples input adv(X, Y) :- facts(X, "hasAdvisor", Y). result(Y) :- adv("Einstein", Y). result(Y) :- result(X), adv(X, Y). π_2 $\pi_{\scriptscriptstyle 3}$ $\sigma_{1=\mathit{Einstein}}$ $\bowtie_{1=1}$ $\pi_{\scriptscriptstyle 1,3}$ $\pi_{1,3}$ $\sigma_{2=hasAdvisor}$ $\sigma_{2=hasAdvisor}$ input input

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Dutline

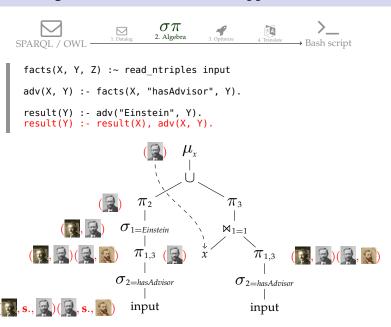
Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization Experiments



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

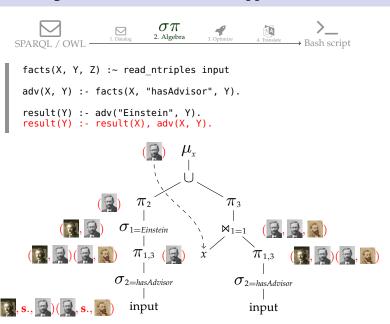
Answering Queries with Unix Shell

Idea

Approach Optimization

Summary

Lonciusion



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

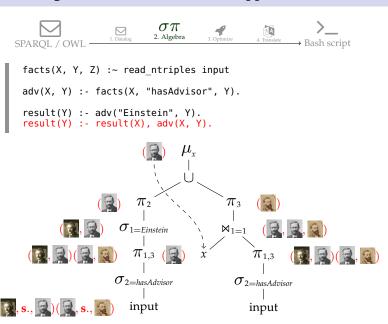
Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

> Experiments Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

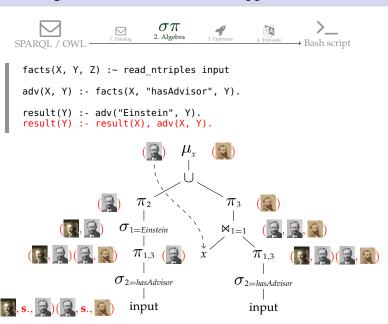
Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization

Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization Experiments

Summary

32/41

 $\sigma\pi$ 2. Algebra SPARQL / OWL → Bash script facts(X, Y, Z) :~ read ntriples input adv(X, Y) :- facts(X, "hasAdvisor", Y). result(Y) :- adv("Einstein", Y). result(Y) :- result(X), adv(X, Y). π_3 π_2 $\sigma_{1=Einstein}$ $\bowtie_{1=1}$ $\pi_{1,3}$ $\pi_{1,3}$ $\sigma_{2-hasAdvisor}$ $\sigma_{2=hasAdvisor}$ input input

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

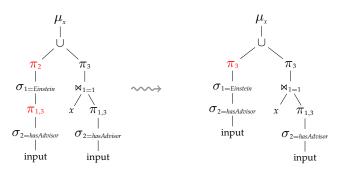
Idea Approach

Optimization

Summary



Optimizations:



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

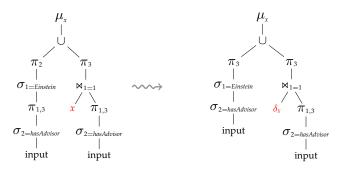
Idea Approach

Optimization Experiments

Summary



Optimizations:



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

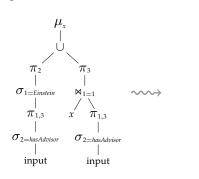
Motivation Idea

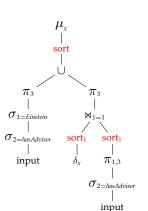
Approach Optimization

Summary



Optimizations:





Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

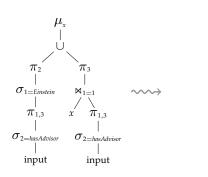
Answering Queries with Unix Shell

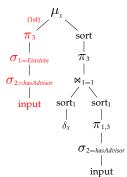
Idea Approach

Optimization Experiments



Optimizations:





Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

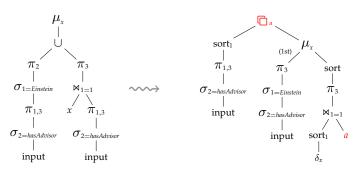
Idea

Approach Optimization

Experiments Summary



Optimizations:



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix

Idea

Approach
Optimization
Experiments



```
awk '
           $1 == "Einstein"
                                                   && $2 == "hasAdvisor")
 sort<sub>1</sub>
                                                     { print $3 >> "b" }
               (1st)
                                                ($2 == "hasAdvisor")
 \pi_{1.3}
               \pi_3
                           sort
                                                     { print $1 FS $3 >> "pre a" }
                                                  ' <(read ntriples input)</pre>
\sigma_{2=adv.}
           \sigma_{1=Einstein}
                           \pi_3
            \sigma_{\scriptscriptstyle 2=adv.}
input
                          \bowtie_{1=1}
                                                 lock a
             input
                       sort<sub>1</sub>
                                                     sort - k 1 pre a > a
                                                    # unlock a
                                                 &
```

Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach
Optimization

Summary

Adding Words to Regexes

Answering Queries with Unit Shell

Idea

Approach
Optimization
Experiments

Experiments Summary

```
awk '
          $1 == "Einstein"
                                                && $2 == "hasAdvisor")
 sort<sub>1</sub>
                                                  { print $3 >> "b" }
              (1st)
                                             ($2 == "hasAdvisor")
 \pi_{1.3}
              \pi_3
                         sort
                                                  { print $1 FS $3 >> "pre a" }
                                               ' <(read ntriples input)</pre>
\sigma_{2=adv}.
          \sigma_{1=Einstein}
                          \pi_3
           \sigma_{2=adv.}
                         \bowtie_{1=1}
                                               lock a
input
             input
                      sort<sub>1</sub>
                                                  sort - k 1 pre a > a
                                                 # unlock a
                                              &
```



```
awk '
           $1 == "Einstein"
                                                  && $2 == "hasAdvisor")
 sort<sub>1</sub>
                                                    { print $3 >> "b" }
                                               ($2 == "hasAdvisor")
 \pi_{1.3}
                          sort
                                                    { print $1 FS $3 >> "pre a" }
                                                 ' <(read ntriples input)</pre>
\sigma_{2=adv.}
           \sigma_{1=Einstein}
                           \pi_3
            \sigma_{\scriptscriptstyle 2=adv.}
input
                          \bowtie_{1=1}
                                                 lock a
             input
                      sort<sub>1</sub>
                                                    sort - k 1 pre a > a
                                                    # unlock a
                                                &
```

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Summary

```
awk '
                                                        $1 == "Einstein"
                                                   && $2 == "hasAdvisor")
 sort<sub>1</sub>
                                                     { print $3 >> "b" }
               (1st)
                                                 ($2 == "hasAdvisor")
 \pi_{1.3}
               \pi_3
                           sort
                                                     { print $1 FS $3 >> "pre a" }
                                                  ' <(read ntriples input)</pre>
\sigma_{2=adv.}
           \sigma_{1=Einstein}
                            \pi_3
            \sigma_{\scriptscriptstyle 2=adv.}
input
                           \bowtie_{1=1}
                                                  lock a
             input
                       sort<sub>1</sub>
                                                     sort - k 1 pre a > a
                                                     # unlock a
                                                 &
```

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Summary

```
awk '
           $1 == "Einstein"
                                                   && $2 == "hasAdvisor")
 sort<sub>1</sub>
                                                    { print $3 >> "b" }
               (1st)
                                                ($2 == "hasAdvisor")
 \pi_{1.3}
               \pi_3
                           sort
                                                    { print $1 FS $3 >> "pre a" }
                                                 ' <(read ntriples input)</pre>
\sigma_{2=adv.}
           \sigma_{1=Einstein}
                           \pi_3
            \sigma_{\scriptscriptstyle 2=adv.}
input
                          \bowtie_{1=1}
                                                 lock a
             input
                       sort<sub>1</sub>
                                                    sort -k 1 pre a > a
                                                    # unlock a
                                                &
```

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Experiments Summary



```
while
 sort<sub>1</sub>
                                                   sort -k 1 -u
                                                         <( # wait for a
 \pi_{1.3}
                \pi_3
                             sort
                                                                join -1 1 -2 1 -0 2.2
                                                                    <(sort -k 1 -u delta)
\sigma_{2=adv.}
            \sigma_{1=Einstein}
                                                                    а
             \sigma_{\scriptscriptstyle 2=adv.}
input
                         sort<sub>1</sub>
              input
                                                       -s delta ];
                                                   do continue; done
```

Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

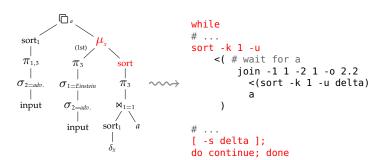
Adding Words to Regexes

Answering Queries with Unit Shell

Idea

Approach
Optimization
Experiments

SPAROL / OWI



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

Bash script

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Idea Approach

Optimization Experiments

Summary



Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Answering

Queries with Uni: Shell

Idea

Optimization

Experiments Summary

Conclusion



How can I find all professors?

Professor(X) :- Person(X),

teachesCourse(X,Y).

Person(X) :- Employee(X).
Person(X) :- Professor(X).



Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Regexes Regexes

Queries with Uni Shell

Idea

Approach Optimization

Experiments

Conclusion



How can I find all professors?

Professor(X) :- Person(X),

teachesCourse(X,Y).

Person(X) :- Employee(X).
Person(X) :- Professor(X).



Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Regexes

Queries with Unit Shell

Idea

Optimization Experiments

Summary

Conclusion

How can I find all professors?

Professor(Y).

Person(X) :- Employee(X).
Person(X) :- Professor(X).

Combining the first and the last rule leads to



Rebele

The YAGO knowledge base

Outline

Using YAGO for the humanities

Regexes Answering

Queries with Unit Shell

Idea

Optimization

Experiments Summary

Conclusion

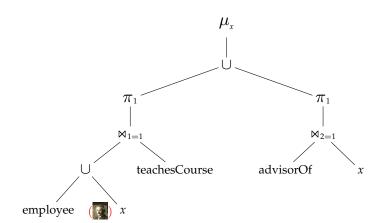
How can I find all professors?

Professor(X) :- Person(X),

teachesCourse(X,Y).
advisorOf(X Y)

Person(X) :- Employee(X).
Person(X) :- Professor(X).

Combining the first and the last rule leads to



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

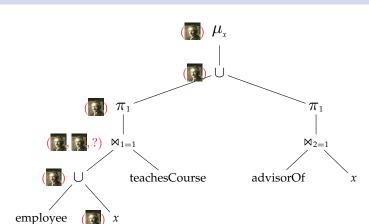
Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Approach Optimization

Experiments Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

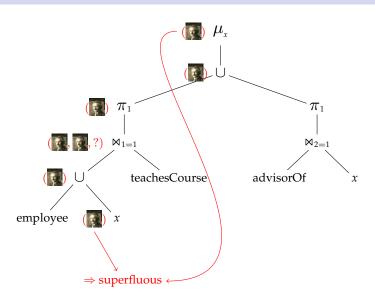
Adding Words to Regexes

Answering Queries with Unix Shell

Idea

Optimization

Experiments Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

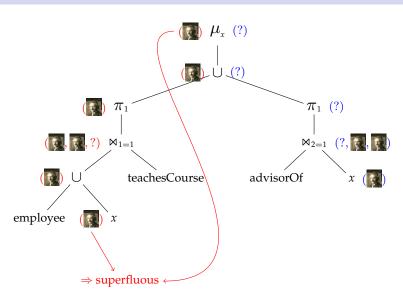
Adding Words to Regexes

Answering Queries with Unix Shell

Motivation Idea

Approach Optimization

Experiments Summary



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outline

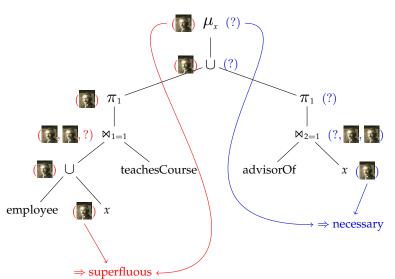
Using YAGO f

Adding Words to Regexes

Answering Queries with Unix Shell

Motivation Idea

Optimization Experiments



Expanding the YAGO knowledge base

Rebele

The YAGO knowledge base

Outlin

Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

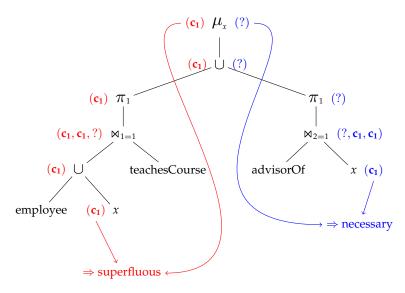
Motivation

Approach Optimization

Experiments Summary

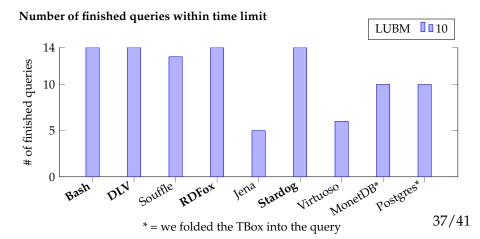


Optimization

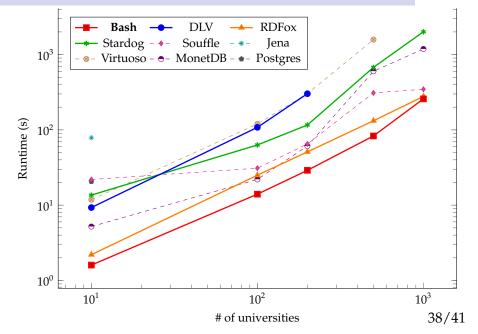


Answering Queries with Unix Shell: Experiments

- ► Dataset: LUBM university benchmark
- ▶ 14 different queries
- Competitors: Datalog-based systems (DLV, Souffle, RDFox),
 Triple stores (Jena, Stardog, Virtuoso),
 Database management systems (MonetDB, Postgres)



Answering Queries with Unix Shell: Experiments



Using YAGO for the humanities

Adding Words to Regexes

Answering Queries with Unix Shell

Motivation Idea

Approach Optimization

Experiments

ummary

nclusion

dataset	Bash	RDFox	BigDatalog	Stardog	Virtuoso
LiveJournal	117	70	532	941	-
orkut	225	121	1838	1123	-
friendster	16306	-	-	-	=

Table: Runtime for the reachability query, in seconds.

Answering Queries with Unix Shell: Summary

Expanding the YAGO knowledge base

Rehele

Summary

Summary:

- Preprocess large datasets without installing software
- Supports subset of SPARQL / OWL and Datalog as query language
- Try it online at https://www.thomasrebele.org/projects/bashlog
- Source code available at https://github.com/thomasrebele/bashlog

Future work:

- Numerical comparisons
- Aggregations (e.g., max, count)

Publication: ISWC 2018 (full paper)







Conclusion

Expanding the YAGO knowledge base

Rebele

he YAGO nowledge base

Dutline

Jsing YAGO fo he humanities

Answering

hell

Conclusion

This thesis showed how to extend YAGO along several axes:

- ____
- ► Improve completeness w.r.t. people
- ► Automatically repairing of its regular expressions
- Preprocessing queries using only a Bash shell

Other accomplishments:

- ▶ Source code of all contributions is available online
- Publications at ISWC 2016 (resource paper), ISWC 2017 (demo, workshop), PAKDD 2018 (full paper), ISWC 2018 (full paper)

















Future work:

- More studies on human society using facts from YAGO (ongoing)
- ► Combine YAGO and Wikidata
- Queries with numerical comparisons and aggregations