

Scalable Factoid QA: Big Knowledge Bases and Complex Questions

Luke Zettlemoyer

joint work with
Yoav Artzi, Eunsol Choi, Oren Etzioni,
Anthony Fader, and Tom Kwiatkowski



UNIVERSITY *of*
WASHINGTON

Question Answering (QA)

Semantic Parsing

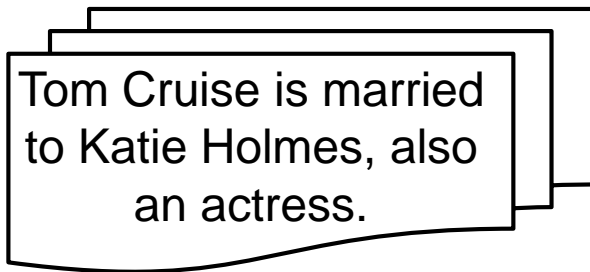
How many states
have a higher point
than the highest point
of the state with the
largest capital city in
the US?



Zelle & Mooney, '96,
Zettlemoyer & Collins, '05,
Liang et al., '11,
and many many more

Answer Retrieval

Who is Tom Cruise
married to?



Voorhees & Tice, '00
Ravichandran & Hovy '02
and many many more

Semantic Parsing

Q: How many people live in Seattle?

Semantic Parser

MR: $\lambda x. \text{population}(\text{seattle}, x)$



620,778

Two Big Challenges

- Part 1: How do we understand complex questions against large, varied KBs?

[Kwiatkowski et al, 2013]

- Part II: How do we get enough facts to answer any question?

[Fader et al, 2014]

Open Domain QA

Q Who managed Liverpool F.C. from 2004 to june 2010?

A Rafael Benitez

Q What architectural style is the Brooklyn Bridge?

A Gothic Revival architecture

Q What are the symptoms of prostate cancer?

A {Hematuria, Nocturia, Dysuria, ... }

Q How many people ride the monorail in Seattle daily?


A 7,000

Open Domain QA

Q

Who managed Liverpool F.C. from 2004 to June 2010?

A

 **Freebase** is a community authored knowledge base with:

Q

A

- 40 Million Entities
- 2 Billion Facts

Q

A

- 20,000 Relations
- 10,000 Types

Q

A

7,000

- 100 Domains

Query is Domain Dependent

How many people live in Seattle?


Query is Domain Dependent

How many people live in Seattle?



$\lambda x. \text{eq}(x, \text{count}(\lambda y. \text{person}(y) \wedge \text{home}(y, \text{seattle})))$

Person	Home
Eunsol	Seattle
Luke	Seattle
Jane	Boston



Query is Domain Dependent

How many people live in Seattle?




$\lambda x. \text{eq}(x, \text{count}(\lambda y. \text{person}(y) \wedge \text{home}(y, \text{seattle})))$



$\lambda x. \text{population}(\text{seattle}, x)$

City	Population
Seattle	620778
Boston	636479



Person	Home
Eunsol	Seattle
Luke	Seattle
Jane	Boston



Query is Domain Dependent


How many people live in Seattle?

- Requires different syntax for different domains
 - Grammars do not generalize well
 - Grammars are hard to learn

 λη.ποπυλασιον(σεαυεις, κ)


Jane Boston

City	Population
Seattle	620778
Boston	636479




Query is Domain Dependent


How many people live in Seattle?

 $\lambda x. \text{population}(\text{seattle}, x)$


City	Population
Seattle	450,000
Boston	750,000



How many people have won the Nobel peace prize?

 $\lambda x. \text{eq}(x, \text{count}(\lambda y. \text{person}(y) \wedge \text{won}(y, \text{nobel_peace_prize})))$

Person	Award
Nelson M.	Nobel P.P.
Mother T.	Nobel P.P.
Leymah G.	Nobel P.P.




Query is Domain Dependent


How many people live in Seattle?

 $\lambda x. \text{population}(\text{seattle}, x)$


City	Population
Seattle	620778
Boston	636479



How many people have won the Nobel peace prize?

 $\lambda x. \text{eq}(x, \text{count}(\lambda y. \text{person}(y) \wedge \text{won}(y, \text{nobel_peace_prize})))$

Person	Award
Nelson M.	Nobel P.P.
Mother T.	Nobel P.P.
Leymah G.	Nobel P.P.



2 Stage Semantic Parsing

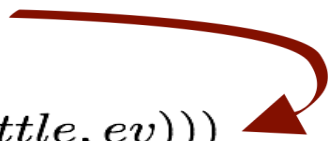
How many people live in Seattle?

2 Stage Semantic Parsing

1. Domain independent, linguistically motivated parse.

How many people live in Seattle?

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists ev.live(y, ev) \wedge in(seattle, ev)))$



2 Stage Semantic Parsing

2. Domain specific ontology match.

How many people live in Seattle?

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists ev.live(y, ev) \wedge in(seattle, ev)))$



$\lambda x.eq(x, count(\lambda y.person(y) \wedge home(y, seattle)))$

Person	Home
Eunsol	Seattle
Luke	Seattle
Jane	Boston

A stack of four white, circular disks, likely representing data storage or a database.

2 Stage Semantic Parsing

2. Domain specific ontology match.

How many people live in Seattle?

$\lambda x. eq(x, count(\lambda y. people(y) \wedge \exists ev. live(y, ev) \wedge in(seattle, ev)))$




$\lambda x. eq(x, count(\lambda y. person(y) \wedge home(y, seattle)))$




$\lambda x. population(seattle, x)$

City	Population
Seattle	620778
Boston	636479



Person	Home
Eunsol	Seattle
Luke	Seattle
Jane	Boston



2 Stage Semantic Parsing


2. Domain specific ontology match.

How many people live in Seattle?


- All domains use same syntax that generalizes well
- Ontology match can be guided by the structure of the underspecified logical form

$\lambda x. \text{population}(\text{Seattle}, x)$

City	Population
Seattle	620778
Boston	636479



Luke	Seattle
Jane	Boston

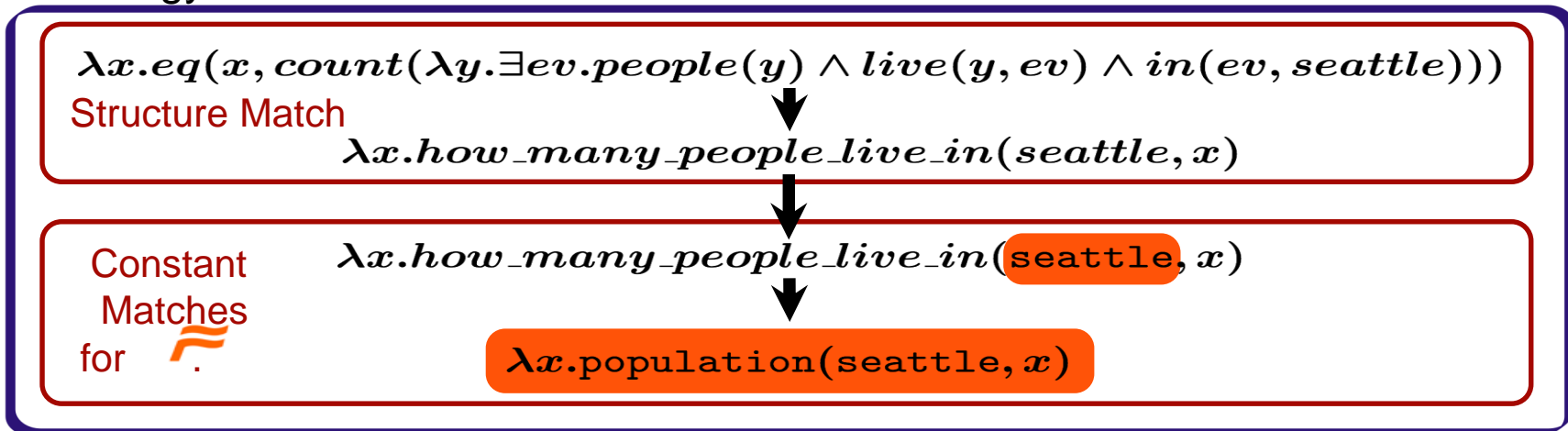


2 Stage Semantic Parsing

Domain Independent Parse

How many	people	live	in	Seattle
$S/(S \setminus NP)/N$	N	$S \setminus NP$	$S \setminus S/NP$	NP
$\lambda f \lambda g \lambda x. eq(x, count(\lambda y. g(y) \wedge f(y)))$	$\lambda x. people(x)$	$\lambda x \lambda ev. live(x, ev)$	$\lambda x \lambda f \exists ev. in(ev, x) \wedge f(ev)$	$seattle$
>				<B
$\lambda x. eq(x, count(\lambda y. \exists ev. people(y) \wedge live(y, ev) \wedge in(ev, seattle)))$				

Ontology Match



Domain Independent Parsing

49 domain independent lexical items:

Word	Syntax	Underspecified semantics
How many	$\vdash S/(S \setminus NP)/N$	$: \lambda f \lambda g \lambda x. eq(x, count(\lambda y. f(y) \wedge g(y)))$
What	$\vdash S/(S \setminus NP)/N$	$: \lambda f \lambda g \lambda x. f(x) \wedge g(x)$
most	$\vdash NP/N$	$: \lambda f. max_count(\lambda y. f(y))$
etc.		

56 underspecified lexical categories:

Part-of-Speech	Syntax	Underspecified semantics
proper noun	$\vdash NP$	$: C$
noun	$\vdash N$	$: \lambda x. P(x)$
noun	$\vdash N/N$	$: \lambda f \lambda x. f(x) \wedge P(x)$
verb	$\vdash S \setminus NP$	$: \lambda \lambda ev. P(x, ev)$
verb	$\vdash S \setminus NP/NP$	$: \lambda x \lambda y \lambda ev. P(y, x, ev)$
preposition	$\vdash N \setminus N/NP$	$: \lambda f \lambda x \lambda y. P(y, x) \wedge f(y)$
preposition	$\vdash S \setminus S/NP$	$: \lambda f \lambda x \exists ev. P(ev, x) \wedge f(ev)$
etc.		

2 Step Semantic Parsing

Domain Independent Parse

How many	people	live	in	Seattle
$S/(S \setminus NP)/N$	N	$S \setminus NP$	$S \setminus S/NP$	NP
$\lambda f \lambda g \lambda x. eq(x, count(\lambda y. g(y) \wedge f(y)))$	$\lambda x. people(x)$	$\lambda x \lambda ev. live(x, ev)$	$\lambda x \lambda f \exists ev. in(ev, x) \wedge f(ev)$	$seattle$
>				<B
$\lambda x. eq(x, count(\lambda y. \exists ev. people(y) \wedge live(y, ev) \wedge in(ev, seattle)))$				

Ontology Match

$\lambda x. eq(x, count(\lambda y. \exists ev. people(y) \wedge live(y, ev) \wedge in(ev, seattle)))$

Structure Match

$\lambda x. how_many_people_live_in(seattle, x)$

Constant Matches

for



$\lambda x. how_many_people_live_in(seattle, x)$

$\lambda x. population(seattle, x)$

Structural Match

Collapse and expand subexpressions in underspecified logical form with operators that:

1. Collapse simple typed sub-expression
2. Collapse complex typed sub-expression
3. Expand predicate

New example

How many people ride the monorail in Seattle daily?

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, \iota z.monorail(z) \wedge in(z, seattle), e) \wedge daily(e)))$



`$\lambda x.transit_system/daily_riders(seattle_monorail, x)$`

Structural Match

1. Find subexpression with type allowed in KB

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, \iota z.monorail(z) \wedge in(z, seattle), e) \wedge daily(e)))$

2. Replace with new underspecified constant

Structural Match

1. Find subexpression with type allowed in KB

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, \lambda z.monorail(z) \wedge in(z, seattle), e) \wedge daily(e))))$

entity typed
subexpression

2. Replace with new underspecified constant

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, monorail_in_seattle), e) \wedge daily(e)))$

Structural Match

1. Find subexpression with type allowed in KB

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, \iota z.monorail(z) \wedge in(z, seattle), e) \wedge daily(e)))$

integer typed
subexpression

2. Replace with new underspecified constant

$\lambda x.eq(x, count(\lambda y.people(y) \wedge \exists e.ride(y, monorail_in_seattle), e) \wedge daily(e)))$

$\lambda x.eq(x, how_many_people_ride_daily_the_monorail_in_seattle)$

Constant Match

Replace constants with constants from KB

$\lambda x.$ how_many_people Ride_daily(the_monorail_in_seattle, x)

Assume constants have English string labels!

Constant Match

Replace constants with constants from KB

$\lambda x.$ how_many_people_ride_daily(the_monorail_in_seattle, x)



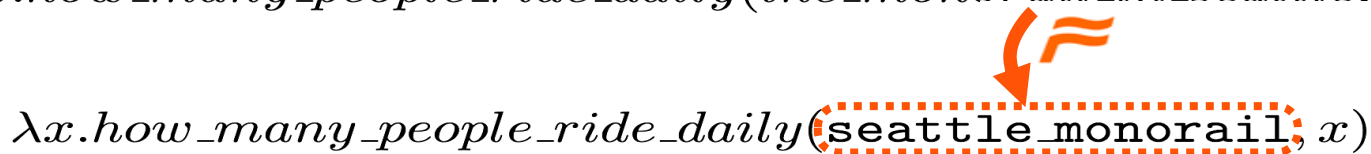
`$\lambda x.$ transit_system/daily_riders(seattle_monorail, x)`


Constant Match

Replace constants with constants from KB

$\lambda x.$ how_many_people_ride_daily(the_monorail_in_seattle, x)

$\lambda x.$ how_many_people_ride_daily(seattle_monorail, x)



 $\lambda x.$ transit_system/daily_riders(seattle_monorail, x)

Constant Match

Replace constants with constants from KB

$\lambda x. \text{how_many_people_ride_daily}(\text{the_monorail_in_seattle}, x)$

$\lambda x. \text{how_many_people_ride_daily}(\text{seattle_monorail}, x)$

$\lambda x. \text{transit_system/daily_riders}(\text{seattle_monorail}, x)$

$\approx \lambda x. \text{transit_system/daily_riders}(\text{seattle_monorail}, x)$

2 Stage Semantic Parsing

Domain Independent Parse

How many	people	live	in	Seattle
$S/(S \setminus NP)/N$	N	$S \setminus NP$	$S \setminus S/NP$	NP
$\lambda f \lambda g \lambda x. eq(x, count(\lambda y. g(y) \wedge f(y)))$	$\lambda x. people(x)$	$\lambda x \lambda ev. live(x, ev)$	$\lambda x \lambda f \exists ev. in(ev, x) \wedge f(ev)$	<i>seattle</i>
>		<		
$\lambda x. eq(x, count(\lambda y. \exists ev. people(y) \wedge live(y, ev) \wedge in(ev, seattle)))$				

Ontology Match

$\lambda x. eq(x, count(\lambda y. \exists ev. people(y) \wedge live(y, ev) \wedge in(ev, seattle)))$

Structure Match

$\lambda x. how_many_people_live_in(seattle, x)$

Constant Matches for

$\lambda x. how_many_people_live_in(seattle, x)$

$\lambda x. population(seattle, x)$

Learning

Input

Q/A pairs $\{(x_i, a_i) : i = 1, \dots, n\}$

Knowledge Base, Wiktionary, Underspecified Lexicon

Algorithm

For $i = 1, \dots, n$

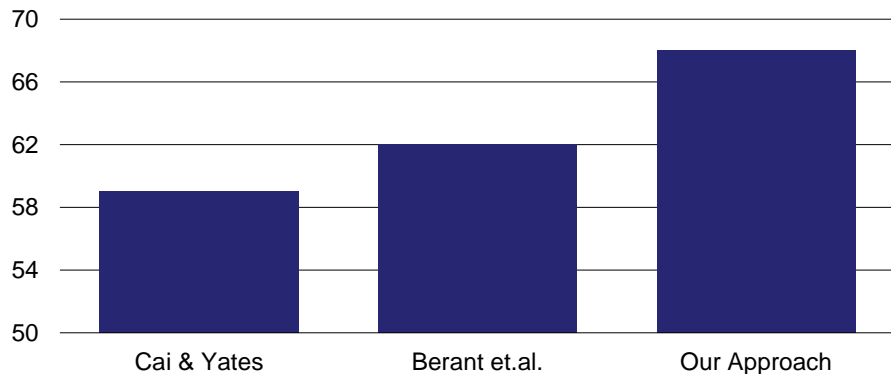
$C \leftarrow$ Max scoring correct parses of x_i

$W \leftarrow$ Margin violating incorrect parses of x_i

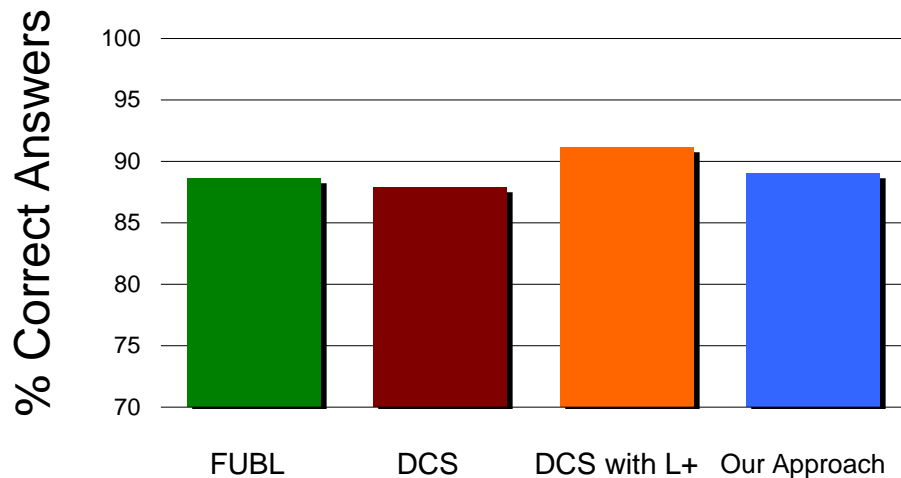
$$\theta = \theta + \frac{1}{|C|} \sum_{c \in C} \phi(c) - \frac{1}{|W|} \sum_{w \in W} \phi(w)$$

Experiments

Works well on
Freebase benchmark



Also works on older,
more compositional
benchmark



Example Parses

How many operating systems is Adobe Flash compatible with?



```
 $\lambda x. \text{eq}(x, \text{count}(\lambda y. \text{software\_compatibility} \\ \text{.operating\_system}(\text{adobe\_flash}, y))))$ 
```

Who is the CEO of Save-A-Lot?



```
 $\lambda x. \text{person}(x) \wedge \exists y. \text{organization}(y, \text{savealot}) \wedge \\ \text{board\_member.leader\_of}(x, y) \wedge \text{leadership.role}(y, \text{ceo})$ 
```


Example Errors

How many children does Jerry Seinfeld have?

Target:

 $\lambda x.\text{eq}(x, \text{count}(\lambda y.\text{person.children}(\text{jerry_seinfeld}, y)))$

Prediction:

 $\lambda x.\text{eq}(x, \text{count}(\lambda y.\text{person.children}(y, \text{jerry_seinfeld})))$

Example Errors

What programming languages were used for AOL instant messenger?

Target:



```
 $\lambda x. \text{languages\_used}(\text{aol\_instant\_messenger}, x)$ 
```

Prediction:



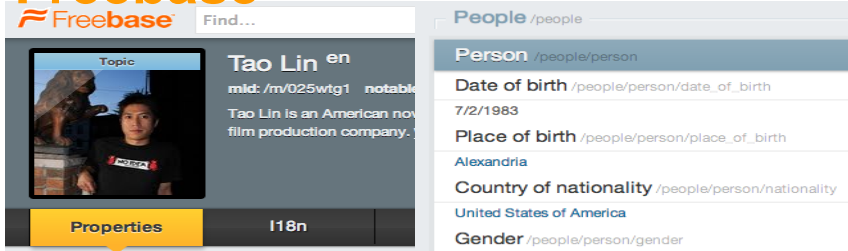
```
 $\lambda x. \text{languages\_used}(\text{aol\_instant\_messenger}, x)$   
 $\wedge \text{programming\_language}(x)$ 
```

Two Big Challenges

- Part 1: How do we understand complex questions against large, varied KBs?
 - Use underspecified semantic parser
 - Learn to match meaning to target domain
- Part II: How do we get enough facts to answer any question?

Open Information Extraction (ReVerb)

Freebase



Freebase Find... People /people

Person /people/person

Date of birth /people/person/date_of_birth
7/2/1983

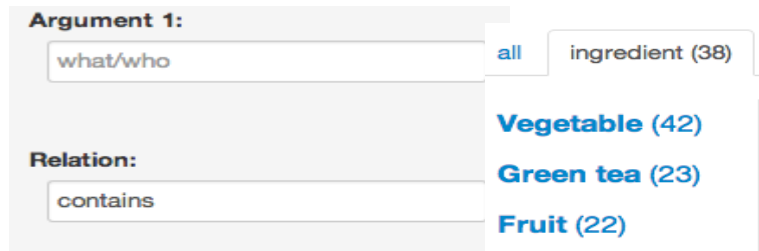
Place of birth /people/person/place_of_birth
Alexandria

Country of nationality /people/person/nationality
United States of America

Gender /people/person/gender

Topic: Tao Lin en
mid: /m/025wtg1 notable
Tao Lin is an American novelist and film production company.

Properties 118n



Argument 1:
what/who all ingredient (38)

Relation:
contains Vegetable (42)
Green tea (23)
Fruit (22)

**Multiple Knowledge Sources
Billions of Facts
Millions of Relationships**

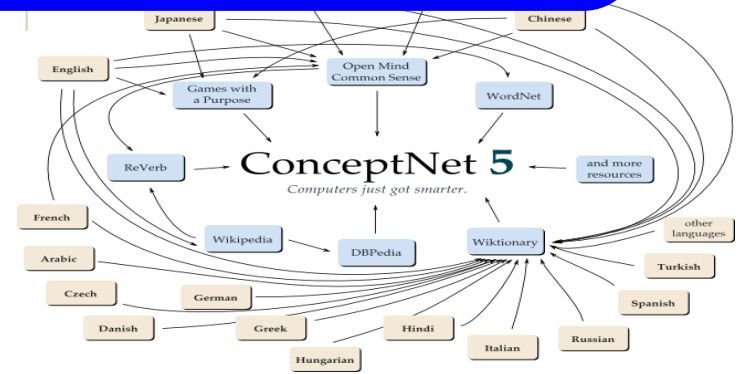
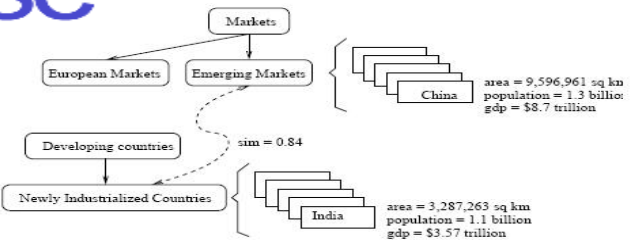
Read

Research Pro

Home

NELL: Never-Ending Language Learning

ProBase

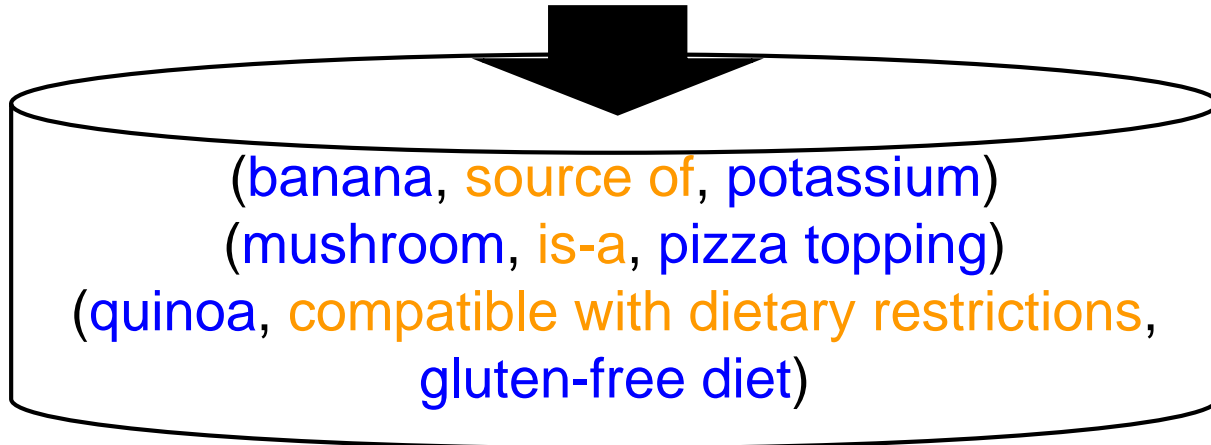


Knowledge Base



Open IE

ProBase

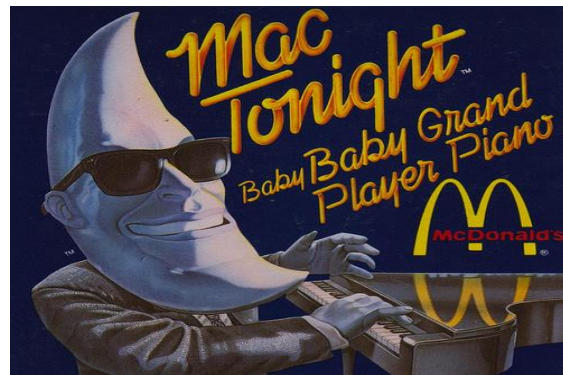


Don't require:
Normalization
Canonicalization
Ontologization

Freebase Triples

Required

arg1: McDonald's
rel: Advertising characters
arg2: Mac Tonight
namespace: Freebase



Optional

arg1 id: /m/0jg57
arg2 id: /m/01qq3s

Threw away
non-binary
relations

arg1: McDonald's
rel: Revenue
arg2: **Currency:** United States Dollar
Amount: 23,000,000,000
Valid Date: 12/31/2008



Open IE Triples

Bill Clinton ate at McDonalds, went jogging, and confessed a fondness for trashy spy novels.

Required

arg1: Bill Clinton
rel: ate at
arg2: McDonald's
namespace: Open IE



Optional

confidence: 0.823

frequency: 3

arg1 id: /m/0157m

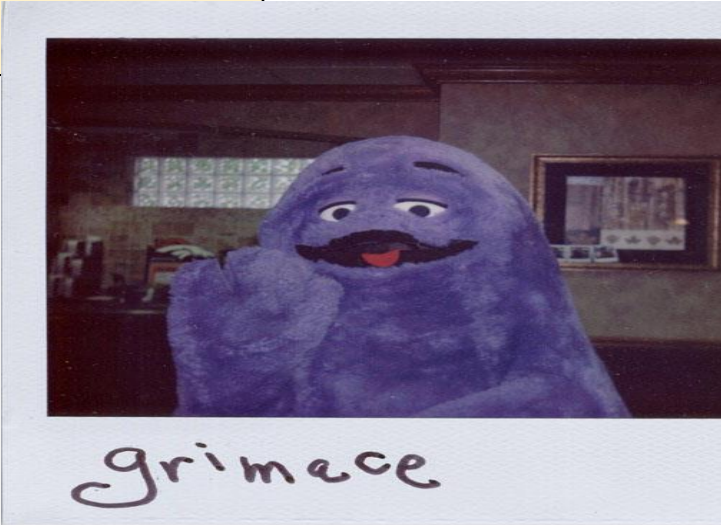
arg2 id: /m/07gyp7

Lin et al. 2012

Probase Triples (Wu et al., 2012)

McDonald's characters such as Grimace or Mac Tonight...

Required	arg1: Grimace
	rel: is a
	arg2: McDonald's characters
	namespace: Probase
Optional	arg1 frequency: 17
	arg2 frequency: 4
	popularity: 1
	arg1 vagueness: 0.67
	...



NELL Triples (Carlson et al., 2010)

Sammy Sosa, who played for the Cubs in...
Cubs player Sammy Sosa came from the...
Last night Sammy Sosa hit a run for the Cubs...

Required {
Optional {

arg1:	sammy sosa
rel:	plays for the team
arg2:	cubs
namespace:	nell
<hr/>	
confidence:	0.92

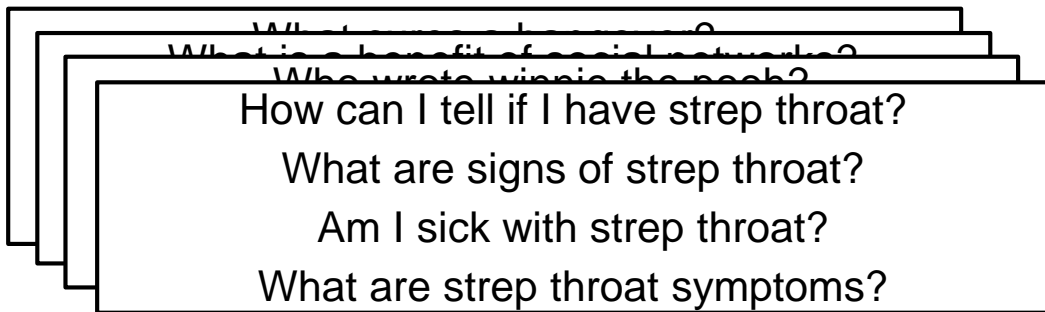


Sammy Sosa,
1998

Knowledge Base Statistics

Source	Type	# Triples	# Relation Phrases
Freebase	Curated	300M	18K
Open IE	Extracted	500M	6M
Probase	Extracted	200M	1
NELL	Extracted	2M	300

WikiAnswers Paraphrase Corpus



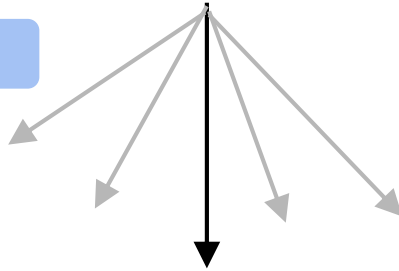
20 million user-created
clusters

First used for Open QA in Paralex
(Fader, Zettlemoyer, and Etzioni, 2012)

How can I tell if I have the flu?

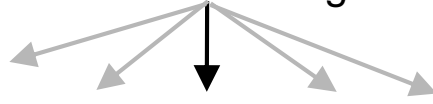
(flu, symptoms, the chills)

Paraphrase



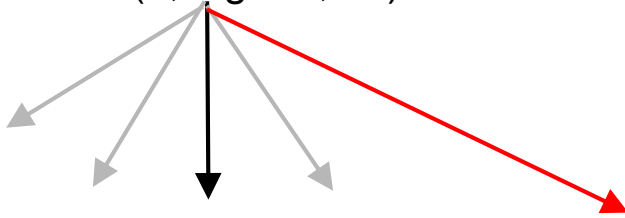
What are signs of the flu?

Parse



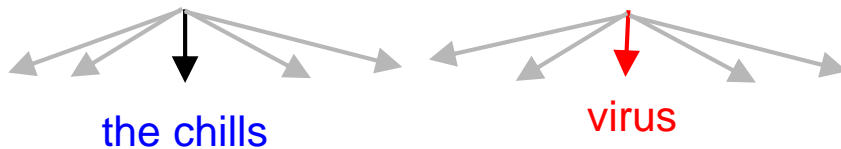
(?, sign of, flu)

Rewrite



(flu, symptoms, ?) (?, causes, flu)

Execute



the chills

virus



How can I tell if I have strep throat?
 What are signs of strep throat?
 Am I sick with strep throat?



sign of → (cough, cold) (fever, strep)
 ...
 symptoms⁻¹ → (cough, cold) (fever, strep) ...

score(... → the chills) = 1.2
 score(... → virus) = -0.3

Query Language

What fruits are a source of vitamin C?

?x: (?x, is a, fruit) (?x, source of, vitamin c)

arg1	rel	arg2
Lychee	is a	fruit
star-fruit	is a	tropical fruit
pepper	is a	fresh fruit



arg1	rel	arg2
Lychees	good source of	vitamin C
starfruits	source of	vitamin C
peppers	provides a source of	vitamins C and A

Mining Paraphrase Templates

How can I tell if I have **strep throat**?
What are signs of **strep throat**?
Am I sick with **strep throat**?
What are **strep throat** symptoms?

t_1	t_2	count(t_1, t_2)	count(t_1)	count(t_2)
How can I tell if I have _	What are signs of _	8	21	45
How can I tell if I have _	Am I sick with _	11	21	52
How can I tell if I have strep _	What are signs of strep _	1	1	1

5 million $\{t_1, t_2\}$ pairs with $\text{count}(t_1, t_2) \geq 5$

Cherry-Picked Paraphrases

Template 1

How does _ affect your body?

What is the latin name for _?

Why do we use _?

What to use instead of _?

Was _ ever married?

Template 2

What body system does _ affect?

What is _'s scientific name?

What did _ replace?

What is a substitute for _?

Who has _ been married to?

Mining Query Rewrite Operators

arg1	arg2	sign of	symptom ⁻¹
cough	cold	✓	✓
jealousy	love	✓	
dizziness	Meniere's	✓	✓
chills	flu		✓

74 million $\{r_1, r_2\}$ pairs with ≥ 10 shared arguments

DIRT (Lin and Pantel, 2001)

Sempre (Berant et al., 2013)

Cherry-Picked Rewrite Rules

Source Relation

Target Relation

(?x, children, ?y)

(?y, was born to, ?x)

(?x, birthdate, ?y)

(?x, date of birth, ?y)

(?x, headquartered in, ?y)

(?y, is based in, ?x)

(?x, invented, ?y)

(?y, was invented by, ?x)

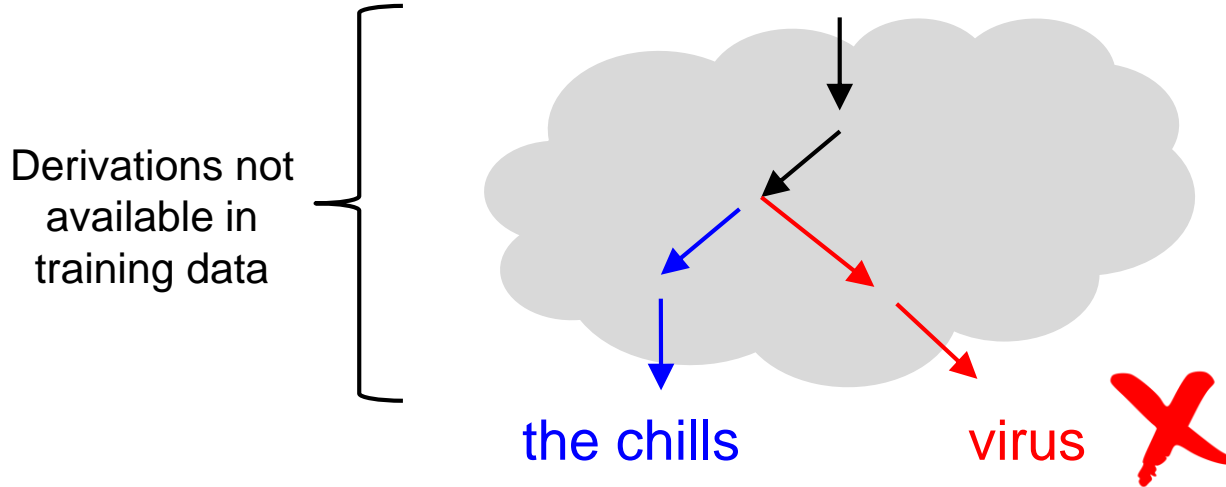
(?x, is the language of, ?y)

(?y, languages spoken, ?x)

Learning and Inference

Latent-Variable Structured Perceptron: Liang et al., 2006; Sun et al., 2009

How can I tell if I have the flu?



$$\mathbf{w} = \mathbf{w} + \mathbf{f}(\dots \rightarrow \text{the chills}) - \mathbf{f}(\dots \rightarrow \text{virus})$$

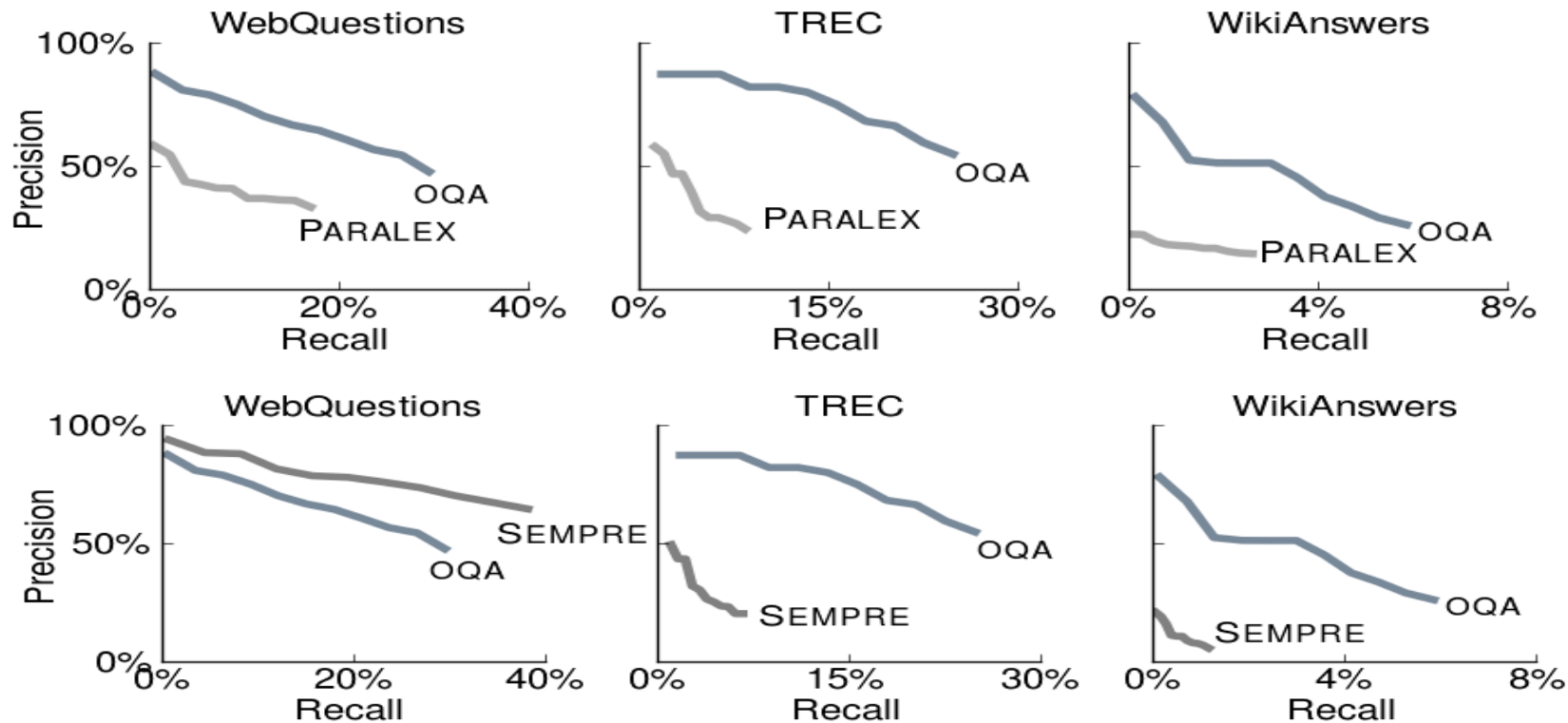
Question Sets

WebQuestions Berant et al., 2013
where was nicki minaj born?

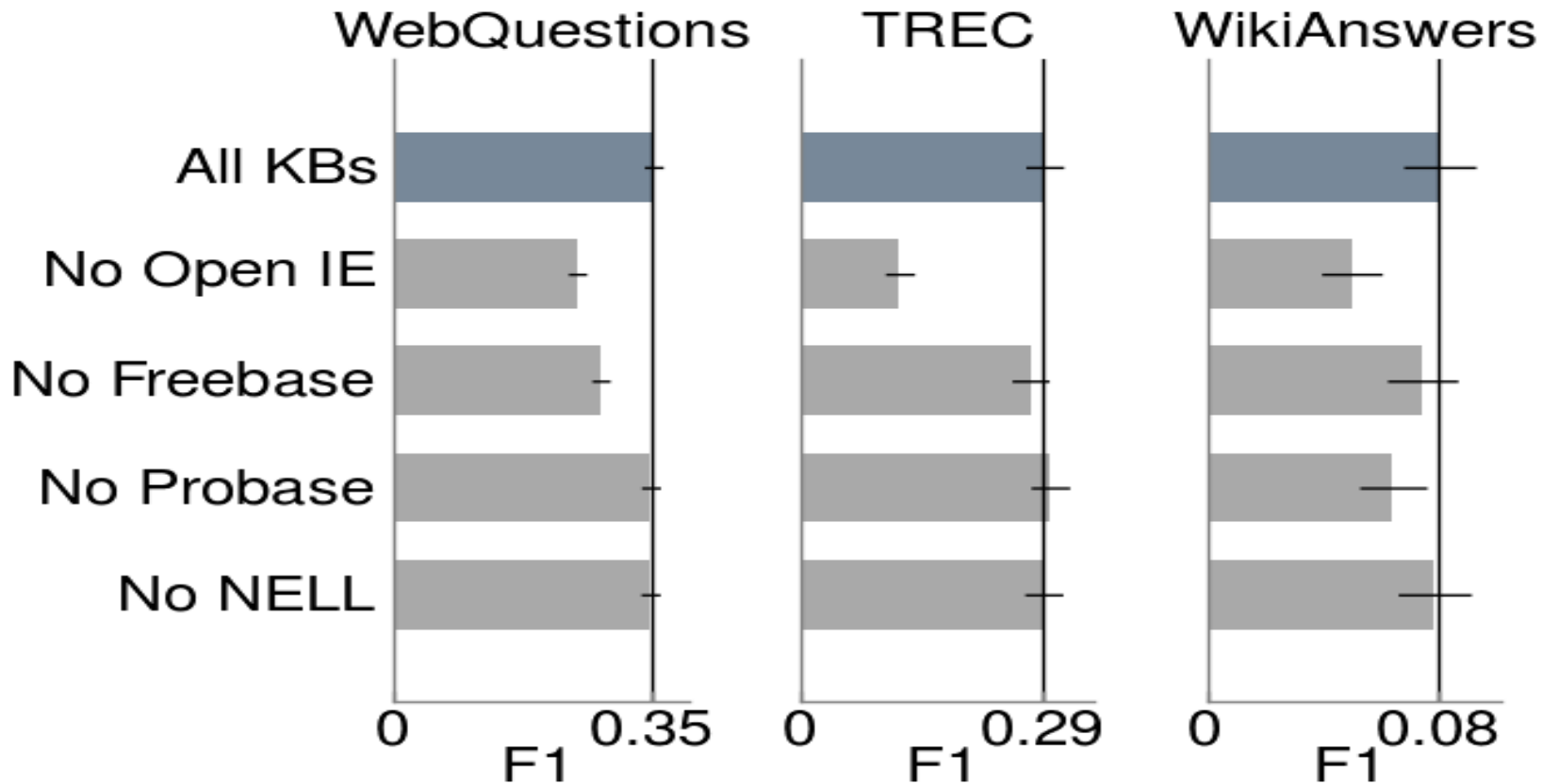
TREC Vorhees and Tice, 2000
What other countries do curds live in ?

WikiAnswers Held-out from corpus
Who is a retired gay nfl player?

Experiments



Effect of Knowledge Sources



Cherry-Picked Examples

Input	Who did Michael J Fox marry?
Parse	?x: (Michael J Fox, marry, ?x)
Rewrite	?x: (Michael J Fox, has wife, ?x)
Execute	Tracy Pollan: (Michael J. Fox, has wife, Tracy Pollan)

Input	What are brake pads made of?
Paraphrase	What material are brake pads made of?
Parse	?x: (?x, is-a, material) (brake pads, made of, ?x)
Execute	copper: (copper, is-a, material) (The brake pads, were made of, copper)

Lemon-Picked Examples

Input	What animal represents California?
Paraphrase	What are California's symbols?
Parse	?x: (california, symbols, ?x)
Execute	CWT: (California Water Service, Trading symbol, CWT)
Input	What actor first portrayed James Bond?
Parse	?x: (?x, is-a, actor first) (?x, portrayed, James Bond)
Execute	Daniel Craig: (Daniel Craig, is-a, first class actor) (Danny Craig, portrays, James Bond)

An Embarrassingly Funny Example

What kind of sound does a unicorn make?

What does a unicorn

(a really cute, fun)

fun



Two Big Challenges

- Part 1: How do we understand complex questions against large, varied KBs?
 - Use underspecified semantic parser
 - Learn to match meaning to target domain
- Part II: How do we get enough facts to answer any question?
 - Use large, unstructured tuple store
 - Learn from paraphrases and KB facts

A Few Open Questions

- How many facts do we need, and can we ever enumerate them all?
- Can we jointly learn to extract facts and answer questions?
- Can we apply similar techniques to understand non-question sentences?