Learning to Execute Natural Language

Percy Liang

Computer Forum
April 16, 2013

Why natural language?

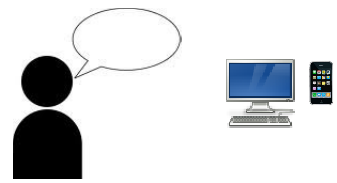
Information extraction

Observation: free-form text contains a wealth of information (e.g., news articles, blogs, scientific journals, etc.)

Goal: extract structured information from unstructured natural language

---

Intelligent user interfaces

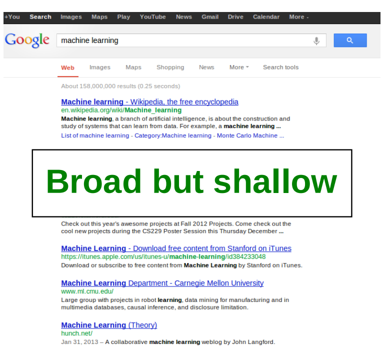


Tasks: find information, perform data analytics, buy tickets, send email, etc.

Goal: allow people to accomplish more complex tasks efficiently using natural language

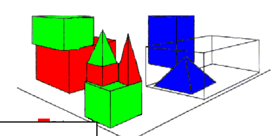
Book a non-stop flight from SF to Tokyo when the price drops by 20%.

Web search



Broad but shallow

Blocks world [Winograd 1971]



Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don't understand which pyramid you mean.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: By "it", I assume you mean the block which is taller than the one I am hol
Computer: OK.
Person: What does the box
Computer: The blue pyrami
Person: What is the pyramid supported by?
Computer: The box.
Person: How many blocks are not in the box?
Computer: Four of them.
Person: Is at least one of them narrower than the one which I told you to pick up?
Computer: Yes, the red cube.

Deep but narrow

*How do we get **deep** and **broad** systems?*

Models in NLP

Of countries that don't border an ocean, which has the most people?

Basic models:

- Topic models (e.g., Latent Dirichlet Allocation)
- n-gram language models
- Sequence models (e.g., HMM, conditional random fields)

More structured models (our focus):

- Syntactic models over parse trees
- Semantic models over logical forms

Deep question answering

Of countries that don't border an ocean, which has the most people?

$\xrightarrow{\text{semantic parsing}}$

$$\mathbf{argmax}(\lambda x. \mathbf{Country}(x) \wedge \neg\exists y. \mathbf{Border}(x, y) \wedge \mathbf{Ocean}(y), \lambda x. \mathbf{Population}(x))$$

$\xrightarrow{\text{execute database query}}$

`Egypt`

Point: to answer question, need to model the **logical form**

Training a semantic parser

Detailed supervision: manually annotate logical forms

What's Bulgaria's capital?	$\mathbf{Capital}(\mathbf{Bulgaria})$
When was Google started?	$\mathbf{DateFounded}(\mathbf{Google})$
What movies has Tom Cruise been in?	$\lambda x. \mathbf{Movie}(x) \wedge \mathbf{ActedIn}(\mathbf{TomCruise}, x)$
...	...

Requires experts — slow and expensive, doesn't scale up!

Example: Penn Treebank (50K sentences annotated with parse trees) took 3 years

Training a semantic parser

Shallow supervision: question/answers pairs

What's Bulgaria's capital?	`Sofia`
When was Google started?	`1998`
What movies has Tom Cruise been in?	`TopGun,VanillaSky,...`
...	...

- Get answers via crowdsourcing (no expertise required) or by scraping the web — fast and cheap (but noisy), scales up
- Logical forms modeled as **latent variables**

Summary so far:

- Modeling **deep** semantics of natural language is important
- Need to learn from natural/weak supervision to obtain **broad** coverage

Rest of talk:

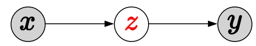
- Spectral methods for learning latent-variable models
- Learning a broad coverage semantic parser

Spectral methods for learning latent-variable models

(joint work with Daniel Hsu, Sham Kakade, Arun Chaganty)

Latent-variable models

natural/weak supervision $\Rightarrow$ latent variables

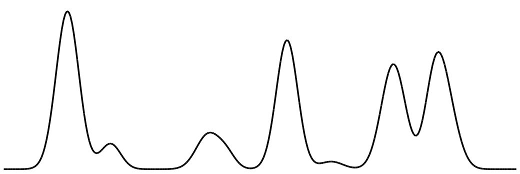


Many applications:

- Semantic parsing
- Relation extraction
- Machine translation
- Speech recognition
- ...

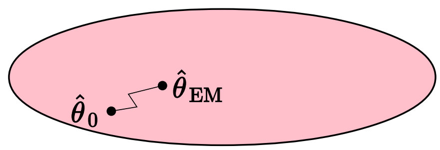
Unsupervised learning

In general, latent-variable models lead to non-convex optimization problems (finding global optimum is NP hard)



Local optimization

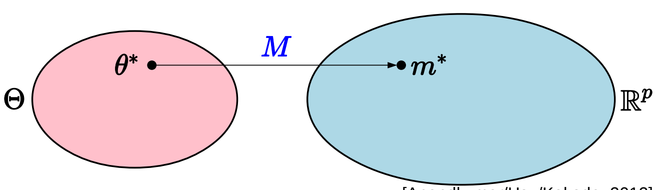
Algorithms: EM, Gibbs sampling, variational methods



Problem: get stuck in local optima

Solution (**heuristic**): careful initialization, annealing, multiple restarts

Method of moments (global)



[Anandkumar/Hsu/Kakade, 2012]

Algorithm (has rigorous theoretical guarantees):

- Compute aggregate statistics over data (trivial to parallelize)
- Perform simple linear algebra operations to obtain parameter estimates

Method of moments (global)

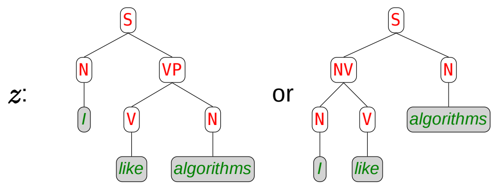
	Use of data	Computation
Global optimization	efficient	inefficient
Local optimization	no guarantees	
Method of moments	inefficient	efficient

In Big Data regime, method of moments is a win!

Missing: structural uncertainty, discriminative modeling

Structural uncertainty

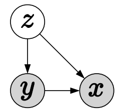
x: *I like algorithms.*



Our algorithm: unmixing [NIPS 2012]

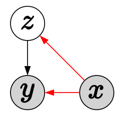
Discriminative latent-variable models

Generative models (e.g., Naive Bayes):



$$p(x, y, z)$$

Discriminative models (e.g., logistic regression, SVMs):



$$p(y, z \mid x)$$

Our algorithm: for mixture of linear regressions [ICML 2013]

Semantic parsing

(joint work with Jonathan Berant, Andrew Chou)

Semantic parsing

Of countries that don't border an ocean, which has the most people?

semantic parsing

$$\mathbf{argmax}(\lambda x.\,\mathbf{Country}(x) \wedge \neg\exists y.\,\mathbf{Border}(x, y) \wedge \mathbf{Ocean}(y), \lambda x.\,\mathbf{Population}(x))$$

execute database query

`Egypt`

Training data

Expensive: logical forms

[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005]

[Wong & Mooney, 2006; Kwiatkowski et al., 2010]

What is the most populous city in California?
$\Rightarrow \mathbf{argmax}(\lambda x.\,\mathbf{city}(x) \wedge \mathbf{loc}(x, \mathbf{CA}), \lambda x.\,\mathbf{pop}.(x))$

How many states border Oregon?
$\Rightarrow \mathbf{count}(\lambda x.\,\mathbf{state}(x) \wedge \mathbf{border}(x, \mathbf{OR})$

Cheap: answers

[Clarke et al., 2010]

[Liang et al., 2011]

What is the most populous city in California?
$\Rightarrow$ *Los Angeles*

How many states border Oregon?
$\Rightarrow$ *3*

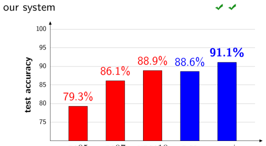
Can we learn with no annotated logical forms?

Experimental results

Task: US geography question/answering benchmark

On Geo, 600 training examples, 280 test examples

System	Description	Lexicon	Logical forms
zc05	CCG [Zettlemoyer & Collins, 2005]	✗ ✗	✓
zc07	relaxed CCG [Zettlemoyer & Collins, 2007]	✗ ✗	✓
kzgs10	CCG w/unification [Kwiatkowski et al., 2010]	✗ ✗	✓
dcs	our system	✓ ✗	✗
dcs$^+$	our system	✓ ✓	✗



Punchline: our system (without logical forms) matches previous work (with logical forms)

Towards broad coverage

Collecting question answering dataset from the Web:

What shows has David Spade been in?
What are the main rivers in Egypt?
What year did Vince Young get drafted?
In what year was President Kennedy shot?
...

Compared to previous datasets:

- Domain: from US geography to general facts
- Database size: from 500 to 400,000,000 (Freebase)
- Number of database predicates: from 40 to 30,000

Alignment

Challenge: figure out how words (e.g., *born*) map onto predicates (e.g., `PlaceOfBirth`)

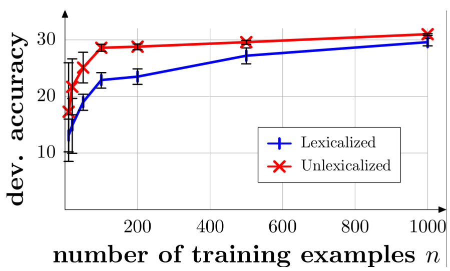
Raw text: 1B web pages   Freebase: 400M assertions

grew up in	`DateOfBirth`
born in	`PlaceOfBirth`
married in	`Marriage.StartDate`
born in	`PlacesLived.Location`

Output: noisy mapping from words to predicates

Final step: train semantic parser using this mapping

Experimental results



Punchline: using alignment, can get same accuracy with 10 times fewer question/answer pairs

Summary

- Goal: **deep** natural language semantics from **shallow** supervision
- Consequence: need to learn **latent-variable models**
- Spectral methods: from intractable to easy by trading off computation and information — paradigm shift in learning
- Semantic parsing: state-of-the-art results learning only from question-answer pairs

Real-world impact

Increasing demand for **deep** language understanding

Thank you!