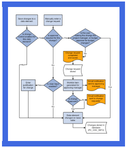
Latent-Variable Grammars and Natural Language Semantics

Shay Cohen
School of Informatics
University of Edinburgh

May 4, 2016

Modern Natural Language Processing



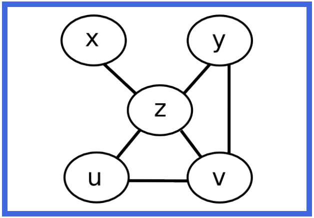
1980s - rule based systems

1990s - statistical methods (frequentist?)

2000s - ... statistical methods (Bayesian analysis?)

2010s - continuous representations, deep learning

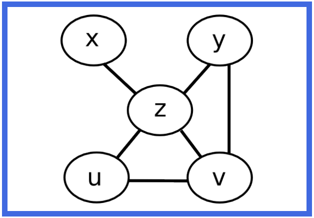
Grammar Models



: Machine Learning

$\boxed{?}$: NLP

Grammar Models



:

Machine Learning

S $\rightarrow$ NP VP
DT $\rightarrow$ the
...

:

NLP

Problems in Semantics Broadly Construed

Any problem in NLP that requires "understanding" text at some level

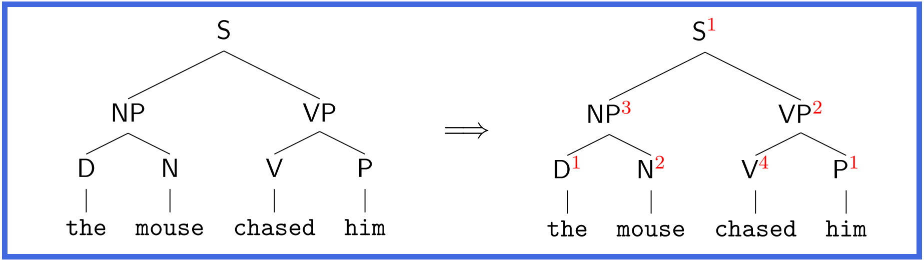
Most prominently: requires lexical understanding (such as in lexical semantics)

By negation: anything that is not syntax!
(or, goes beyond syntax?)

Syntax in Pictures

Semantics in Pictures

PCFGs with Latent States



- Latent states play the role of syntactic heads, as in lexicalization
- They are not part of the observed data in the treebank

This Talk: L-PCFGs for Several Problems

We used latent-variable PCFGs to make progress on several problems:

- Syntactic parsing (the classic application of L-PCFGs)
- Machine translation
- Analyzing social media forums
- Open-domain question answering

Latent-Variable PCFGs

- The probability of a tree is the product of rules with latent states:

$$p(t) = \prod p(a(h_1) \rightarrow b(h_2)\, c(h_3) \mid a(h_1))$$

- It is just a PCFG!

- However, we are interested in distributions over "skeletal" trees

$$p(\text{skeleton}(t)) = \sum_h \prod p(a(h_1) \rightarrow b(h_2)\, c(h_3) \mid a(h_1))$$

- Distributions over skeletal trees are more expressive than PCFG

Main Estimation Algorithm

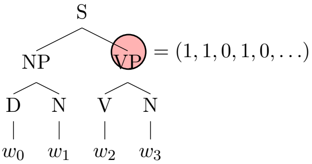
- Some variant of spectral learning for L-PCFGs (ACL, 2012)
- Based on the method of moments, with intuitive-to-understand variants (EMNLP, 2015)
- Performs pretty well on syntactic parsing problems
- Very (computationally) efficient
- Allows encoding information about the latent states directly (why is this important?)

Four Variants of the Estimation Algorithm

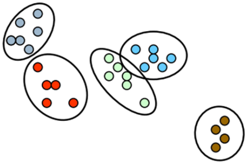
- **SVD variant**: based on singular value decomposition of empirical count matrices (ACL, 2012; JMLR 2014)
- **Convex EM variant**: based on the so-called "anchor method" that identifies features that uniquely identify latent states (ACL, 2014)
- **Clustering variant**: a simplified version of the SVD variant that clusters low-dimensional representations to latent states (EMNLP, 2015)
- **Simpler clustering variant**: a further simplified version of the clustering variant (EMNLP, 2015b)

The (Simpler) Clustering Variant

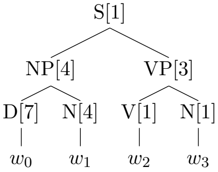
1. For each node in a tree, create a context **feature vector**, as a function of the node and the nodes surrounding it



2. Cluster these vectors to m clusters using a clustering algorithm



3. Annotate each node with the cluster ID of its feature vector. Cluster ID = latent state



The Method of Moments

MoM: Set up equations involving moments and parameters

Then, solve with respect to the parameters

Recent use: finding the parameters of latent variables

Simplicity of estimation

Efficiency of estimation

Theoretical guarantees

Performance

Theoretical Results of Convergence

With EM: local maximization of the log-likelihood function

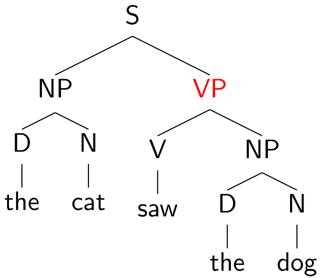
If we could globally maximize the log-likelihood, estimation is likely to be "consistent"

With method of moments: sample complexity statements in the style of:

> *With n samples, the deviation between the estimated probability distribution and the "true" one is small if n is large. The error is a function of various elements, including n, some spectral elements of the moments and properties of the grammar.*

Inside Features Used

Consider the VP node in the following tree:

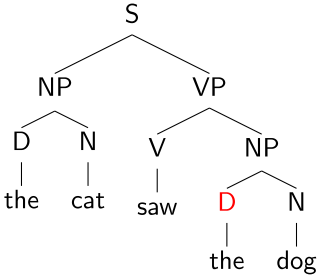


The inside features consist of:

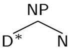
- The pairs (`VP`, `V`) and (`VP`, `NP`)
- The rule `VP` $\rightarrow$ `V NP`
- The tree fragment `(VP (V saw) NP)`
- The tree fragment `(VP V (NP D N))`
- The pair of head part-of-speech tag with VP: (`VP`, `V`)

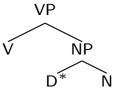
Outside Features Used

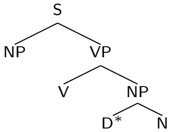
Consider the D node in the following tree:



The outside features consist of:

- The fragments 

, 

and 

- The pair (D, NP) and triplet (D, NP, VP)
- The pair of head part-of-speech tag with D: (D, N)

Results

Out-of-the-box accuracy of the clustering variant:

English: 86.48%
German: 75.04%

Regular spectral algorithm (Cohen et al., 2013):

English: 88.53%
German: 77.71%

Diversity in Parsing

- Models in the clustering variant are very compact
- Idea: create multiple models and combine them together
- Three ways to combine models together
- I will focus on MaxEnt reranking (Charniak and Johnson, 2005)

Creating Multiple Models

- Basic idea: noise the feature representations
- We used dropout and Gaussian noise (Wang et al., 2013)
- Dropout: zero randomly a small fraction of the features
- Gaussian noise: add random Gaussian noise to pre-clustered vectors

Oracle Results with Multiple Models

Oracle results:

English: 95.73%
German: 90.12%

Regular spectral algorithm (Cohen et al., 2013):

English: 92.81%
German: 83.45%

Final Results with Diversity

Final results, MaxEnt reranking:

English: 90.18%
German: 83.38%

Regular spectral algorithm (Cohen et al., 2013):

English: 89.06%
German: 80.64%

Optimizing the Latent State Number

Each nonterminal a is associated with m_a latent states

Spectral learning gives a natural way to choose the number of latent states based on the number of non-zero singular vectors

This criterion does not take into account interactions between different nonterminals

Can we improve that?

The Berkeley parser has proved that coarse-to-fine techniques that carefully select the number of latent states are very useful

Algorithm for Latent State Optimization

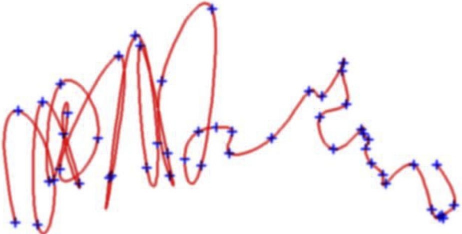
A beam search algorithm

The value in the queue is F_1 measure on a development set

We iterate through the nonterminals, and change the number of latent states, training, calculating accuracy and updating the queue

Major advantage: the training algorithm is relatively fast, so this is mangageable

Traversal of multidimensional vectors (latent state numbers), each giving an F_1 score



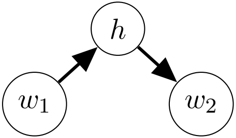
Results

Language	Berkeley	Spectral	Optimized	
Basque	74.7	79.6	80.5	⋆
French	79.9	78.0	78.1	
German	80.1	76.4	79.4	
Hebrew	87.0	86.5	89.0	⋆
Hungarian	85.2	86.5	88.4	⋆
Korean	78.5	76.5	80.0	⋆
Polish	86.7	90.5	91.2	⋆
Swedish	80.6	76.4	79.4	

Experiments: Language Modeling

Saul and Pereira (1997):

$$p(w_2|w_1) = \sum_{h} p(w_2|h)p(h|w_1).$$



This model is a specific case of L-PCFG

Experimented with bi-gram modeling for the Brown corpus and Gigaword corpus

Results: Perplexity

	Brown			NYT		
m	*128*	*256*	test	*128*	*256*	test
bigram Kneser-Ney	408		415	271		279
trigram Kneser-Ney	386		394	150		158
EM	388	365	364	284	265	267
iterations	9	8		35	32	
pivot	426	597	560	782	886	715

Results: Perplexity

	Brown			NYT		
m	*128*	*256*	test	*128*	*256*	test
bigram Kneser-Ney	408		415	271		279
trigram Kneser-Ney	386		394	150		158
EM iterations	388 9	365 8	364	284 35	265 32	267
pivot	426	597	560	782	886	715
pivot+EM iterations	**310** 1	**327** 1	357	**279** 19	292 12	281

- Initialize EM with our algorithm's output
- EM converges in much fewer iterations
- Called "two-step estimation" (Lehmann and Casella, 1998)

This Talk: L-PCFGs for Several Problems

We used latent-variable PCFGs to make progress on several problems:

- Syntactic parsing (the classic application of L-PCFGs)
- Machine translation
- Analyzing social media forums
- Open-domain question answering

Machine Translation

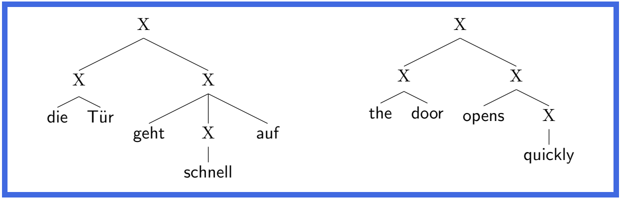
Hierarchical phrase-based MT: use a synchronous grammar

Rewrite pairs of phrases that are mutual translations

There is a single nonterminal X

Example rule: $X \rightarrow \langle X$ good day $| X$ god dag $\rangle$

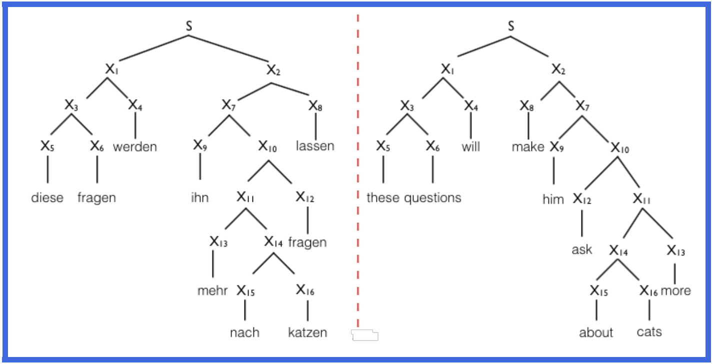
The Need for Context



- Context in translation models is not sufficient for long sentences
- Solution: add latent variables to X

  X is now typed with a category that is not observed in data

Machine Translation



Synchronous L-PCFGs for hierarchical translation (Saluja et al., 2014)

Chinese to English Experiments

System	BLEU score (test)
Hiero	55.3
EM, $m = 8$	49.8
EM, $m = 16$	53.0
Spectral, $m = 8$	53.6
Spectral, $m = 16$	55.8

Spectral grammars are much smaller

They use “minimal grammars” instead of “composed grammars”

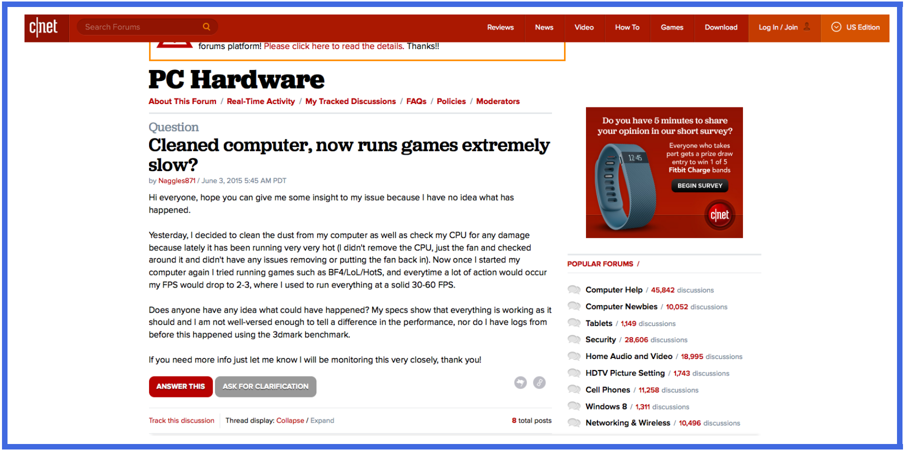
Assume each pair of sentences has a single synchronous derivation

This Talk: L-PCFGs for Several Problems

We used latent-variable PCFGs to make progress on several problems:

- Syntactic parsing (the classic application of L-PCFGs)
- Machine translation
- Analyzing social media forums
- Open-domain question answering

Discussion Forums



cnet | Search Forums | Reviews | News | Video | How To | Games | Download | Log In / Join | US Edition

forums platform! Please click here to read the details. Thanks!!

PC Hardware

About This Forum / Real-Time Activity / My Tracked Discussions / FAQs / Policies / Moderators

Question

Cleaned computer, now runs games extremely slow?

by Naggles871 / June 3, 2015 5:45 AM PDT

Hi everyone, hope you can give me some insight to my issue because I have no idea what has happened.

Yesterday, I decided to clean the dust from my computer as well as check my CPU for any damage because lately it has been running very very hot (I didn't remove the CPU, just the fan and checked around it and didn't have any issues removing or putting the fan back in). Now once I started my computer again I tried running games such as BF4/LoL/HotS, and everytime a lot of action would occur my FPS would drop to 2-3, where I used to run everything at a solid 30-60 FPS.

Does anyone have any idea what could have happened? My specs show that everything is working as it should and I am not well-versed enough to tell a difference in the performance, nor do I have logs from before this happened using the 3dmark benchmark.

If you need more info just let me know I will be monitoring this very closely, thank you!

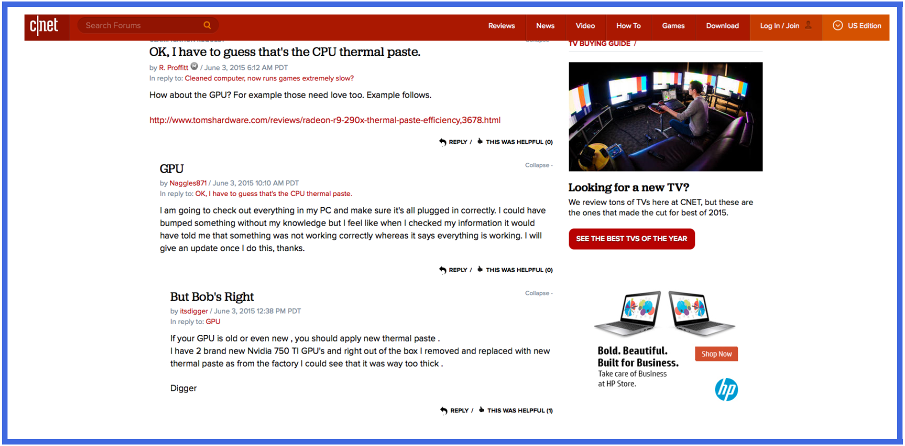
ANSWER THIS | ASK FOR CLARIFICATION

Track this discussion | Thread display: Collapse / Expand | 8 total posts

POPULAR FORUMS /

- Computer Help / 45,842 discussions
- Computer Newbies / 10,052 discussions
- Tablets / 1,149 discussions
- Security / 28,606 discussions
- Home Audio and Video / 18,995 discussions
- HDTV Picture Setting / 1,743 discussions
- Cell Phones / 11,258 discussions
- Windows 8 / 1,311 discussions
- Networking & Wireless / 10,496 discussions

Discussion Forums



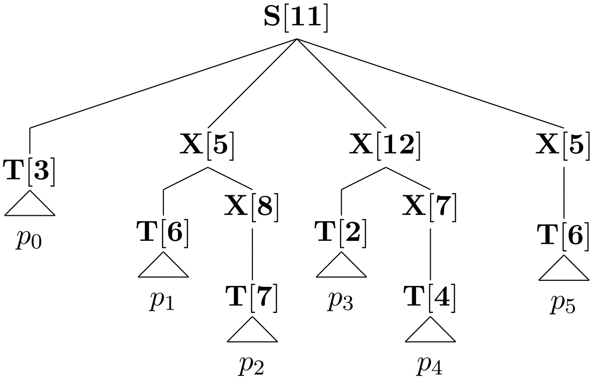
p_0	Bob:	When I play a recorded video on my camera, it looks and sounds fine. On my computer, it plays at a really fast rate and sounds like Alvin and the Chipmunks!
p_1	Kate:	I'd find and install the machine's latest audio driver.
p_2	Mary:	The motherboard supplies the clocks for audio feedback. So update the audio and motherboard drivers.
p_3	Chris:	Another fine mess in audio is volume and speaker settings. You checked these?
p_4	Jane:	Yes, under speaker settings, look for hardware acceleration. Turning it off worked for me.
p_5	Matt:	Audio drivers are at this link. Rather than just audio drivers, I would also just do all drivers.

p_0	Bob:	When I play a recorded video on my camera, it looks and sounds fine. On my computer, it plays at a really fast rate and sounds like Alvin and the Chipmunks!
p_1	Kate:	I'd find and install the machine's latest audio driver.
p_2	Mary:	The motherboard supplies the clocks for audio feedback. So update the audio and motherboard drivers.
p_3	Chris:	Another fine mess in audio is volume and speaker settings. You checked these?
p_4	Jane:	Yes, under speaker settings, look for hardware acceleration. Turning it off worked for me.
p_5	Matt:	Audio drivers are at this link. Rather than just audio drivers, I would also just do all drivers.

p_0	Bob:	When I play a recorded video on my camera, it looks and sounds fine. On my computer, it plays at a really fast rate and sounds like Alvin and the Chipmunks!
p_1	Kate:	I'd find and install the machine's latest audio driver.
p_2	Mary:	The motherboard supplies the clocks for audio feedback. So update the audio and motherboard drivers.
p_3	Chris:	Another fine mess in audio is volume and speaker settings. You checked these?
p_4	Jane:	Yes, under speaker settings, look for hardware acceleration. Turning it off worked for me.
p_5	Matt:	Audio drivers are at this link. Rather than just audio drivers, I would also just do all drivers.

p_0	Bob:	When I play a recorded video on my camera, it looks and sounds fine. On my computer, it plays at a really fast rate and sounds like Alvin and the Chipmunks!
p_1	Kate:	I'd find and install the machine's latest audio driver.
p_2	Mary:	The motherboard supplies the clocks for audio feedback. So update the audio and motherboard drivers.
p_3	Chris:	Another fine mess in audio is volume and speaker settings. You checked these?
p_4	Jane:	Yes, under speaker settings, look for hardware acceleration. Turning it off worked for me.
p_5	Matt:	Audio drivers are at this link. Rather than just audio drivers, I would also just do all drivers.

Conversation Trees



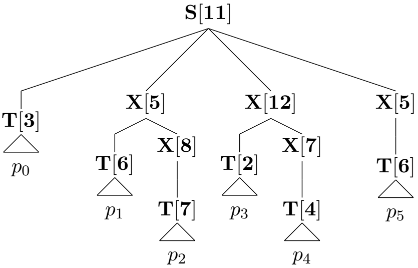
A terminal node is a whole post now!

Conversation Trees

We have grammar rules such as $S[11] \to T[3]\, X[5]\, X[12]\, X[5]$

Each latent state corresponds to a bag of words, a topic

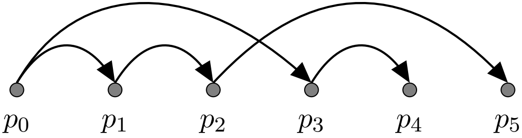
The topic dominates the set of posts in the thread at the bottom



Linear ordering of posts: by time (PCFG not enough!)

Data and Task

- 13,352 threads of computer troubleshooting posts from cnet.com
- Number of posts per thread ranges from 1 to 394
- Threads are structured as dependency trees (reply structure)
- Convert the reply structure to a conversation tree
- Given a set of posts, we try to recover the thread structure

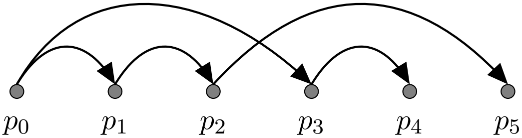


What Features We Use?

- The depth of a node
- Number of siblings
- Number of posts a node dominates
- Bag of words in dominated posts by a node
- ... and others

Further Extensions

- Latent-variable *context-free rewriting systems*
- Allow discontinuous spans for a given nonterminal
- This allows us to have *non-projective* reply structure



Recovery of Conversation Trees

Method	G-p	G-r	NG-p	NG-r	F (final score)
Right branching	35	100	100	0	0
All to root	100	0	56.8	87.6	0
Random	52.2	16.8	54.9	75.1	31.7
PCFG	39.4	58.3	48.8	36.1	**45.0**
PLCFRS	36.3	65.3	55.7	31.6	44.4

Bottom-line: PCFG does the best on average

Number of latent states used: a few dozens per nonterminal

Handling discontinuities does not help that much in total

Recovery of Conversation Trees

Non-projective trees:

PCFG: 43.0%

PLCFRS: 44.1%

LCFRS help with non-projective trees

This Talk: L-PCFGs for Several Problems

We used latent-variable PCFGs to make progress on several problems:

- Syntactic parsing (the classic application of L-PCFGs)
- Machine translation
- Analyzing social media forums
- Open-domain question answering

Open-Domain Question Answering

With the advent of new datasets, open-domain question answering has become a new challenge

Current (industry) systems have relatively high precision, but really low recall

The way a question is asked is very important

What if it were asked in several different ways? What if we asked the same question more than once?

A Simple QA System Wrapper

1. Take an input question q
2. Generate paraphrases for it: $q_1, \ldots, q_m$
3. Use the QA system to get answers a and $a_1, \ldots, a_m$
4. Take a majority vote or another approach to synthesize an answer from $a, a_1, \ldots, a_m$.

L-PCFGs as a Generator

L-PCFGs are a generative model

Each latent state captures a summary of the tree below it

Does the latent state capture enough information to generate a paraphrase of the tree below it?

L-PCFGs as a Generator

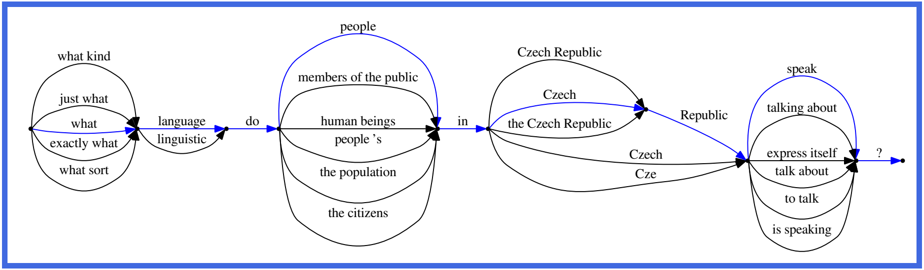
L-PCFGs are a generative model

Each latent state captures a summary of the tree below it

Does the latent state capture enough information to generate a paraphrase of the tree below it?

Unfortunately, the answer is no.

Lattice Constraints



- Constrain the generation to a lattice
- Original question: *what language do people in Czech Republic speak?*

The lattice is created by taking words and phrases from the original sentence together with others from the Paraphrase Database

Generation of Question Paraphrases

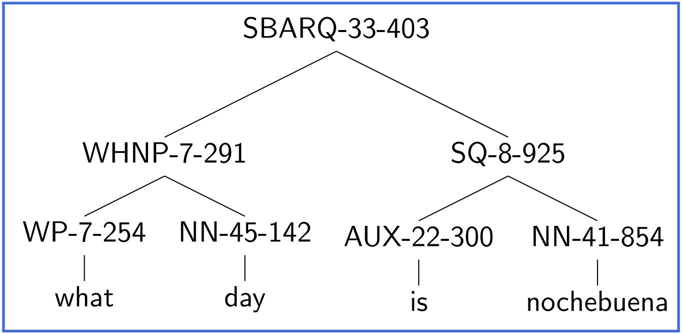
- Paraphrases are generated from an L-PCFG while constraining them to the lattice
- A classifier filters "bad" paraphrases based on simple sentence statistics (such as BLEU score with respect to the original question)
- The classifier is learned from a small amount of manually annotated data with positive and negative examples of paraphrases
- The end result is a black-box that given a question, outputs paraphrases for that question
- The questions are syntactically diverse

How is the L-PCFG Trained?

- We use the Paralex corpus – 18M paraphrase pairs with 2.4M distinct questions
- Parse the questions using the BLLIP parser
- Estimate an L-PCFG with 24 latent states
- How to capture semantic information?

Bi-Layered L-PCFGs

- Add another layer of latent states:



- The second layer has many more latent states (hundreds) and uses a different feature set
- Example feature: bag of words in the inside / outside trees

Semantic Parsing Using Paraphrases

The basic system is from Reddy et al. (2014)

What language do people in Czech Republic speak?

Semantic Parsing Using Paraphrases

The basic system is from Reddy et al. (2014)

What language do people in Czech Republic speak?

What language do people in Czech Republic speak? What is Czech Republic's language? What language do people speak in Czech Republic? ...

Semantic Parsing Using Paraphrases

The basic system is from Reddy et al. (2014)

What language do people in Czech Republic speak?

What language do people in Czech Republic speak? What is Czech Republic's language? What language do people speak in Czech Republic? ...

$\xrightarrow{\text{CCG}}$

$\lambda e.\text{speak.arg1}(e, \text{people})$
$\wedge \text{speak.arg2}(e, \text{language?})$
$\wedge \text{speak.in}(e, \text{CzechRepublic})$

Semantic Parsing Using Paraphrases

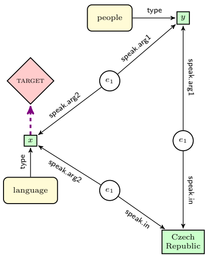
The basic system is from Reddy et al. (2014)

What language do people in Czech Republic speak?

What language do people in Czech Republic speak? What is Czech Republic's language? What language do people speak in Czech Republic? . . .

CCG

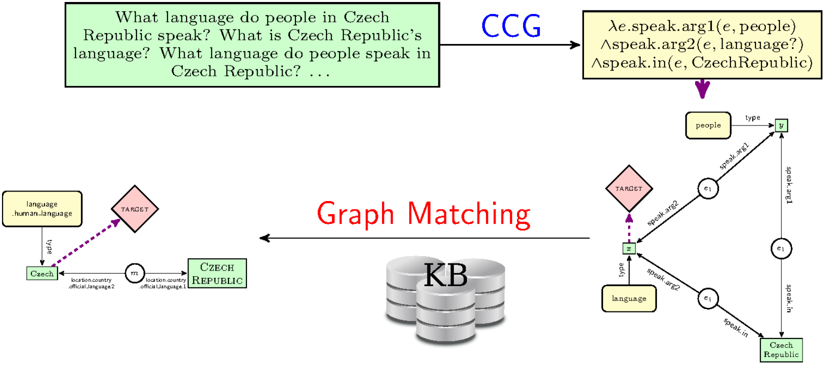
$\lambda e.\text{speak.arg1}(e, \text{people})$
$\wedge \text{speak.arg2}(e, \text{language?})$
$\wedge \text{speak.in}(e, \text{CzechRepublic})$



Semantic Parsing Using Paraphrases

The basic system is from Reddy et al. (2014)

What language do people in Czech Republic speak?



Results

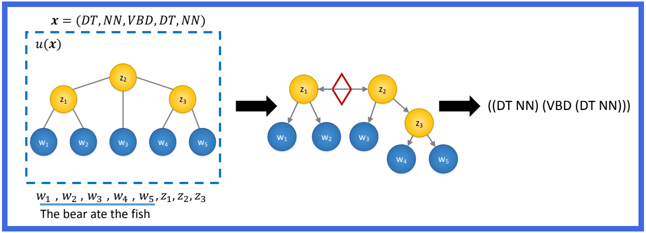
Algorithm	F_1 score
Berant and Liang (2014)	39.9
Bordes et al. (2014)	39.2
Dong et al. (2014)	40.8
Yao (2015)	44.3
Bao et al. (2015)	45.3
Bast and Haussmann (2015)	49.4
Berant and Liang (2015)	49.7
Yih et al. (2015)	52.5
Reddy et al. (2016)	50.3
PPDB	47.7
Bi-layered	48.1

Remarks

- The method does not perform as well as state of the art
- It is partially because paraphrases are done at word level
- Now looking at doing this at a phrase level
- Starting to look more and more like an MT problem

Final Remarks

Another Example: Unsupervised Parsing



Latent structure is a bracketing (Parikh et al., 2014)

Similar in flavor to tree learning algorithms (e.g. Anandkumar, 2011)

Very different flavor from the four estimation algorithms

Another Example: Inference Efficiency

Tensor decomposition for the reduction of grammar constant

Can be used for L-PCFGs (Cohen and Collins, 2012)

Can be used for PCFGs (Cohen et al., 2013)

Turns a chart algorithm into linear in the number of nonterminals

Related algorithm: Rabusseau et al. (2015)

Summary

Summary:

- Latent states + Grammars = Expressive, powerful formalism
- There are theoretically-motivated, efficient ways for estimation
- Various applications