Structured Generative Models for Unsupervised Named Entity Clustering

Micha Elsner, Prof. Eugene Charniak, Prof. Mark E. Johnson

Brown Lab for Linguistic and Information Processing

Brown University BROWN

Providence, RI

Named Entities

People

Micha Elsner
Prof. Eugene Charniak
Prof. Mark E. Johnson

Organizations

Brown Lab for Linguistic and Information Processing
Brown University

Places

Providence, RI

Named Entity Structure

People

	Micha		Elsner
Prof.	Eugene		Charniak
Prof.	Mark	E.	Johnson

Organizations

Brown Lab for Linguistic and Information Processing
Brown University

Places

Providence RI

Motivation

Isn't this old news?

- Cotraining: (Collins+Singer '99, Riloff+Jones '99)

Motivation

Isn't this old news?

- Cotraining: (Collins+Singer '99, Riloff+Jones '99)

Generative models

New direction in coreference resolution:
(Haghighi+Klein '07) (Ng '08) and others
Integrated models for subtasks (including Named Entity)

- (H+K) cluster named entities using...
  - Head word
  - Coreferent pronouns
- Results are promising.
- Can we make them state-of-the-art?

Goal

- Unsupervised, generative model
- Cluster named entities by type

People

Micha Elsner
Prof. Eugene Charniak

Goal

- Unsupervised, generative model
- Cluster named entities by type

People

Micha Elsner
Prof. Eugene Charniak

- Discover word classes

Micha Elsner
Prof. Eugene Charniak

Goal

- Unsupervised, generative model
- Cluster named entities by type

  People

  Micha Elsner
  Prof. Eugene Charniak

- Discover word classes

  Micha Elsner
  Prof. Eugene Charniak

- Cluster possibly-coreferent phrases?

  People

  Micha Elsner
  Prof. Eugene Charniak
  Charniak

Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions

Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

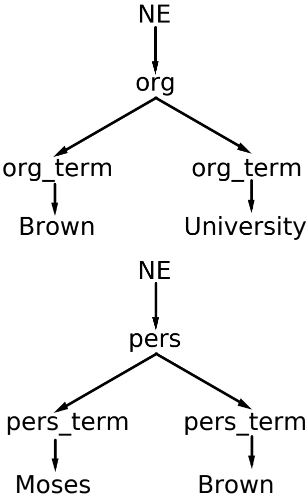
Experiments: pronouns are key!

Future directions

Clustering as parsing

Grammar:

$$NE \rightarrow pers$$
$$NE \rightarrow org$$
$$NE \rightarrow loc$$
$$org \rightarrow org\_term^+$$
$$org\_term \rightarrow \text{Brown}$$
$$org\_term \rightarrow \text{University}$$
$$pers \rightarrow pers\_term^+$$
$$pers\_term \rightarrow \text{Moses}$$
$$pers\_term \rightarrow \text{Brown}$$



Internal structure

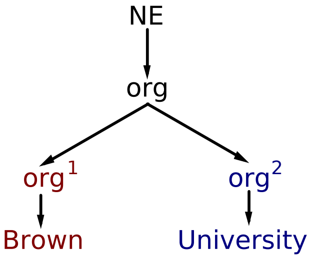
Grammar:

$NE \rightarrow org$

$org \rightarrow org^1 org^2$

$org^1 \rightarrow \text{Brown}$

$org^2 \rightarrow \text{University}$



Internal structure

Grammar:

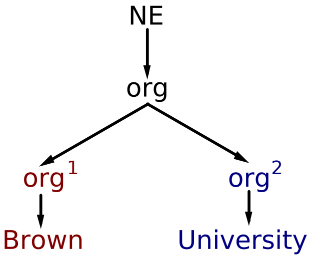
$NE \rightarrow org$

$org \rightarrow org^1 org^2$

$org \rightarrow (org^1)(org^2)(org^3)(org^4)(org^5)$

$org^1 \rightarrow$ Brown

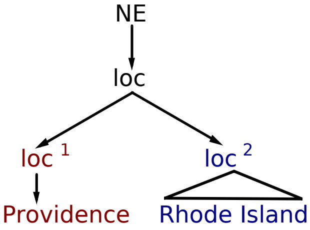
$org^2 \rightarrow$ University



Multiword expansions

Grammar:

$$NE \rightarrow loc$$
$$place \rightarrow loc^1 loc^2$$
$$loc^1 \rightarrow \text{Providence}$$
$$loc^2 \rightarrow \textbf{Rhode Island}$$



Gathering features

- Nominal modifiers (Collins+Singer '99)
  - Appositive: "Hillary Clinton, the **Secretary** of State
  - Prenominal: "**candidate** Hillary Clinton"
- Prepositional governor (C+S '99)
  - "a **spokesman for** Hillary Clinton"
- Personal pronouns
  - "... Hillary Clinton. **She** said ..."
  - Unsupervised model of (Charniak+Elsner '09)
- Relative pronouns
  - "Hillary Clinton, **who** said..."

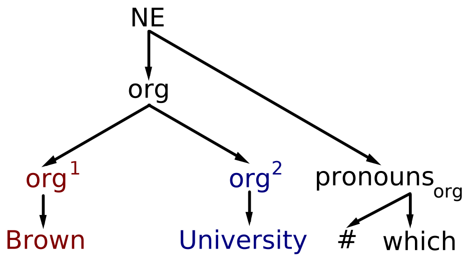
Add features to input strings:

Hillary Clinton # Secretary candidate # spokesman-for # she who

Adding features

Grammar:

$$
\begin{array}{lcl}
NE & \rightarrow & org\ pronouns_{org} \\
org & \rightarrow & org^1 org^2 \\
pronouns_{org} & \rightarrow & \#\ pronoun_{org}{}^* \\
pronoun_{org} & \rightarrow & which \\
pronoun_{org} & \rightarrow & they \\
\ldots & & \\
pronoun_{org} & \rightarrow & he \\
\ldots & &
\end{array}
$$



Learning the grammar

How to learn rule probabilities?

- Many, many rules:
  - With multiword strings, infinite!
- Most of them useless.

Bayesian model

Sparse prior over rules.
Only useful rules get non-zero probability.

Adaptor grammars (Johnson+al '07)

- Prior over grammars
- Form of hierarchical *Dirichlet process*
- Black-box inference, downloadable software
  - Development is just writing the grammar
- But standard inference isn't always good enough

Tuesday, 11:30

"Improving nonparameteric Bayesian inference experiments on unsupervised word segmentation with adaptor grammars", Mark Johnson and Sharon Goldwater.

Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions

Consistent phrases

Definition: Consistent

Phrases that could refer to the same entity.
Weaker than coreference.

Non-trivial for named entities.
Inconsistent, same heads:

- Ford Motor **Co.**
- Lockheed Martin **Co.**

Consistent, different heads:

- Professor **Johnson**
- **Mark**

Modeling consistency

Model's concept of consistency follows (Charniak '01):

Phrases are consistent if none of their internal subparts clash.

Ordered template	pers^1	pers^2	pers^3	pers^4
	Prof.	Mark	E.	Johnson

Modeling consistency

Model's concept of consistency follows (Charniak '01):

Phrases are consistent if none of their internal subparts clash.

Ordered template	pers^1	pers^2	pers^3	pers^4
	Prof.	Mark	E.	Johnson
realizations		Mark		Johnson

Modeling consistency

Model's concept of consistency follows (Charniak '01):

Phrases are consistent if none of their internal subparts clash.

Ordered template	pers^1	pers^2	pers^3	pers^4
	Prof.	Mark	E.	Johnson
realizations		Mark		Johnson
	Prof.			Johnson

Modeling consistency

Model's concept of consistency follows (Charniak '01):

Phrases are consistent if none of their internal subparts clash.

	$pers^1$	$pers^2$	$pers^3$	$pers^4$
Ordered template	Prof.	Mark	E.	Johnson
realizations		Mark		Johnson
	Prof.			Johnson
		Mark		

Modeling consistency

Model's concept of consistency follows (Charniak '01):

Phrases are consistent if none of their internal subparts clash.

	$pers^1$	$pers^2$	$pers^3$	$pers^4$
Ordered template	Prof.	Mark	E.	Johnson
realizations		Mark		Johnson
	Prof.			Johnson
		Mark		
inconsistent		~~Mark~~		~~Steedman~~

Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions

Experimental setup

Datasets:

- Labeled data: MUC-7
  - Three entity classes: PERS, ORG, LOC
- Unlabeled data: NANC

Combine features for multiple examples:

Hillary Clinton #	#	#	who
Hillary Clinton #	Secretary #	#	she
Hillary Clinton #	#	spokesman-for #	her
Hillary Clinton #	Secretary #	spokesman-for #	she her who

More data in equal time...

but no per-document features.

Basic results

Our model:

Baseline (all ORG):	46%
Our best model:	**86%**

Confusion matrix:

	loc	*org*	*per*
LOC	1187	97	37
ORG	223	1517	122
PER	36	20	820

Essentially unjustified comparisons

(Haghighi+Klein ‘07)

- ACE corpus: 61%

(Collins+Singer ‘99)

- Easier dataset
  - Only examples with features
  - Proportionally more people
- Generative baseline: 83%
- Cotraining: 91%

Supervised MUC-7:

- Best system (LTG): 94%
- Human: 97%

Breakdown by features

Model	Dev accuracy
Baseline (All ORG)	42.5
Core NPs (no consistency)	45.5
Core NPs (consistency)	48.5
Context features (nominal/prep)	83.3
All features (context + pronouns)	87.1

Named entity structure

$pers^0$	$pers^1$	$pers^2$	$pers^3$	$pers^4$
rep.	john	minister	brown	jr.
sen.	robert	j.	smith	a
washington	david	john	b	smith
dr.	michael	l.	johnson	iii

loc^0	loc^1	loc^2	loc^3	loc^4
washington	the	texas	county	monday
los angeles	st.	new york	city	thursday
south	new	washington	beach	river
north	national	united states	valley	tuesday

Judging consistency

Sometimes right:

- Dr. Seuss
- Dr. Quinn

... correctly judged inconsistent.

Judging consistency

Sometimes right:

- Dr. Seuss
- Dr. Quinn

... correctly judged inconsistent.

Sometimes wrong:

- Dr. William F. Gibson
- Dr. William Gibson

... judged inconsistent.

- Bruce Jarvis
- Ellen Jarvis

... judged consistent.

Inference is a problem

Gibbs sampling

- Converges in the limit....
- Not in real life!
- Clustering problems are often NP-hard:
  - There's no guaranteed method.

For this model:

- Used heuristic inference
- Still only partial convergence!

Conclusion

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions

Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions

What's next

- Add named-entity to unsupervised coreference
  - Document-level features might help NE...
  - If the combined model could scale.
- Improve inference for Bayesian models
  - Gibbs sampling isn't good enough...
  - Better sampling?
  - Or something completely different?
- Adaptor grammars: what else are they good for?

Thanks!

- Three reviewers
- NSF
- All of you!

Overview

Adaptor grammars: framework for Bayesian grammar learning

Implementing Consistency

Inference: a general problem for this approach

Adaptor grammars (Johnson+al '07)

- A prior over grammars
- Some nonterms are *Dirichlet processes* over subtrees
  - Previously used expansions gain probability
- Black-box inference, downloadable software
  - Development is just writing the grammar
- But standard inference isn't always good enough
  - More on this later...

Tuesday, 11:30

"Improving nonparameteric Bayesian inference experiments on unsupervised word segmentation with adaptor grammars", Mark Johnson and Sharon Goldwater.

Adaptor grammars (Johnson+al '07)

Data:

Prior grammar:

count	*rule*	
1	*words* $\longrightarrow$	*word words*
1	*words* $\longrightarrow$	*word*
1	*word* $\longrightarrow$	Rhode
1	*word* $\longrightarrow$	Island
1	*word* $\longrightarrow$	Colorado
...		
1	$\underline{loc^2}$ $\longrightarrow$	*words*

Providence Rhode Island

Boulder Colorado

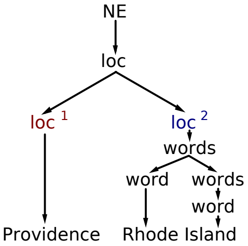
Newport Rhode Island

Adaptor grammars (Johnson+al '07)

Posterior grammar:

count	*rule*	
2	*words* $\longrightarrow$	*word words*
2	*words* $\longrightarrow$	*word*
2	*word* $\longrightarrow$	Rhode
2	*word* $\longrightarrow$	Island
1	*word* $\longrightarrow$	Colorado
...		
1	$\underline{loc^2}$ $\longrightarrow$	*words*
1	$\underline{loc^2}$ $\longrightarrow$	Rhode Island

Data:

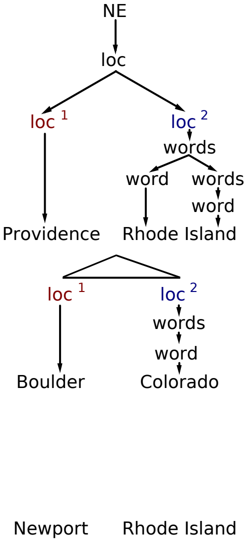


Boulder Colorado

Newport Rhode Island

Adaptor grammars (Johnson+al '07)

Data:



Posterior grammar:

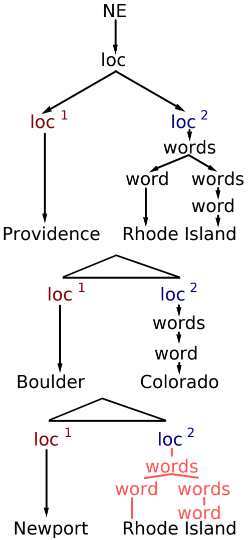
count	*rule*		
2	*words*	$\rightarrow$	*word words*
3	*words*	$\rightarrow$	*word*
2	*word*	$\rightarrow$	Rhode
2	*word*	$\rightarrow$	Island
2	*word*	$\rightarrow$	Colorado
...			
1	$\underline{loc^2}$	$\rightarrow$	*words*
1	$\underline{loc^2}$	$\rightarrow$	Rhode Island
1	$\underline{loc^2}$	$\rightarrow$	Colorado

Adaptor grammars (Johnson+al '07)

Posterior grammar:

count	*rule*	
2	*words* $\rightarrow$	*word words*
3	*words* $\rightarrow$	*word*
2	*word* $\rightarrow$	Rhode
2	*word* $\rightarrow$	Island
2	*word* $\rightarrow$	Colorado
...		
1	$\underline{loc^2}$ $\rightarrow$	*words*
2	$\underline{loc^2}$ $\rightarrow$	Rhode Island
1	$\underline{loc^2}$ $\rightarrow$	Colorado

Data:



Overview

Adaptor grammars: framework for Bayesian grammar learning

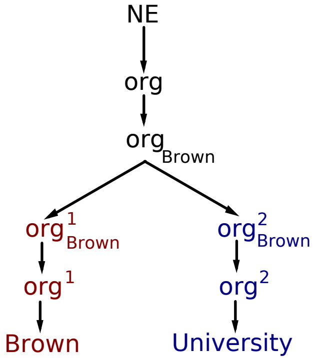
Implementing Consistency

Inference: a general problem for this approach

Implementing consistency

Grammar:

$$
\begin{aligned}
NE &\rightarrow org \\
org &\rightarrow org_{Brown} \cdots \\
org_{Brown} &\rightarrow org^1_{Brown}\ org^2_{Brown} \\
\underline{org^1_{Brown}} &\rightarrow org^1 \\
\underline{org^2_{Brown}} &\rightarrow org^2 \\
\underline{org^1} &\rightarrow \text{Brown} \\
\underline{org^2} &\rightarrow \text{University}
\end{aligned}
$$



Underlined nonterminals are Dirichlet processes.
$\underline{org^1_{Brown}}$ and $\underline{org^2_{Brown}}$ get only one expansion.

Yet another infinity

How many entities (like org_{Brown}) are there?

- Grows with the data size...
- Again, use Bayesian methods.

Allow an infinite number...
and constrain with a sparse prior.

Simple in principle (special case of "Infinite PCFG", Liang+al '07)
Requires some code changes.

Overview

Adaptor grammars: framework for Bayesian grammar learning

Implementing Consistency

Inference: a general problem for this approach

Basic inference by sampling

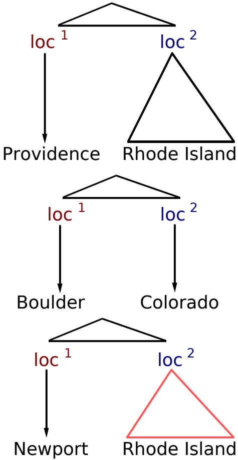
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc}^2$	$\rightarrow$	*words*
1	$\underline{loc}^2$	$\rightarrow$	Colorado
2	$\underline{loc}^2$	$\rightarrow$	Rhode Island

Data:



Basic inference by sampling

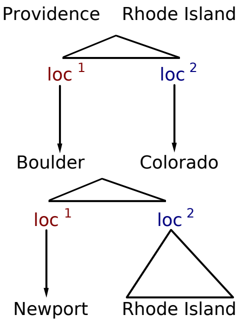
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc^2}$	$\rightarrow$	*words*
1	$\underline{loc^2}$	$\rightarrow$	Colorado
1	$\underline{loc^2}$	$\rightarrow$	Rhode Island

Data:



Basic inference by sampling

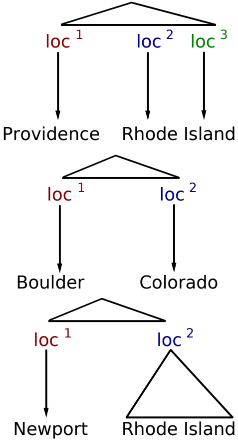
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc}^2$	$\rightarrow$	*words*
1	$\underline{loc}^2$	$\rightarrow$	Colorado
1	$\underline{loc}^2$	$\rightarrow$	Rhode Island

Data:



Basic inference by sampling

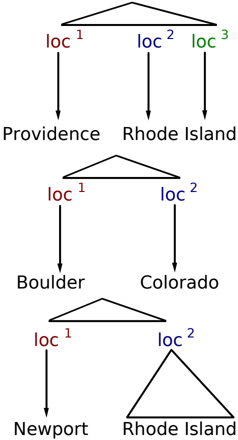
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc}^2$	$\rightarrow$	*words*
1	$\underline{loc}^2$	$\rightarrow$	Colorado
1	$\underline{loc}^2$	$\rightarrow$	Rhode Island
1	$\underline{loc}^2$	$\rightarrow$	Rhode

Data:



Basic inference by sampling

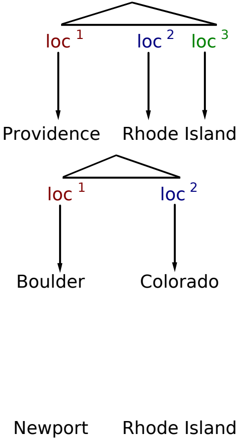
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc}^2$	$\rightarrow$	*words*
1	$\underline{loc}^2$	$\rightarrow$	Colorado
1	$\underline{loc}^2$	$\rightarrow$	Rhode Island
1	$\underline{loc}^2$	$\rightarrow$	Rhode

Data:



Basic inference by sampling

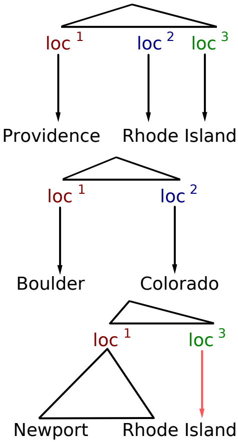
Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

Rules for $\underline{loc}^2$:

1	$\underline{loc}^2$	$\rightarrow$	*words*
1	$\underline{loc}^2$	$\rightarrow$	Colorado
1	$\underline{loc}^2$	$\rightarrow$	Rhode

Data:



Basic inference by sampling

Gibbs sampling:

- Start with arbitrary trees
- Repeat forever
  - Erase a random tree
  - Sample a tree from the current grammar
  - Update the grammar given the new tree

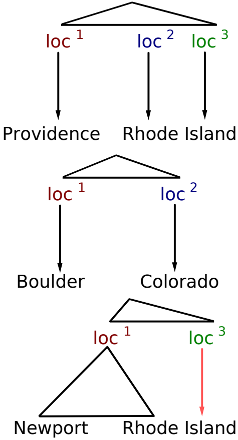
Rules for $\underline{loc}^2$:

$1 \quad \underline{loc}^2 \rightarrow \quad words$

$1 \quad \underline{loc}^2 \rightarrow \quad \text{Colorado}$

$1 \quad \underline{loc}^2 \rightarrow \quad \text{Rhode}$

Data:



Issue 1: efficiency

Sampling a new parse

- Via CKY algorithm: $O(n^3)$
  - ... times a grammar constant!
- One set of nonterminals for each entity
- Scales poorly

Can be dealt with (Metropolis-Hastings algorithm):

- Proposal distribution:
  - Easy-to-calculate approximation to the grammar
- Worse approximations, slower runtimes.

Issue 2: mobility

Local maxima are still a problem

- Gibbs sampling converges in the limit...
- Not in real life!
- What you'd expect – clustering is often NP-hard

- Resampling one tree at a time means lots of local maxima
- Better moves:
  - Split and merge entities
  - Reparse multiple strings at once
- Tricky to implement...
- Correct algorithms can be very slow in practice

Compromise: heuristic inference

What we actually do:

- Propose only a subset of entities for each string:
  - Must have at least one word in common
  - Less likely if shared word is frequent
- *Ignore* the Hastings correction term!

Not theoretically valid, but faster.

- Even so, inference remains a problem.
  - Too many clusters for the same entity