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 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
Prof. Eugene
Prof. Mark E.
Elsner
Charniak
Johnson

Organizations

Brown Lab for Linguistic and Information Processing
Brown University

Places

Providence RI
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:

\[\begin{align*}
NE & \rightarrow \text{pers} \\
NE & \rightarrow \text{org} \\
NE & \rightarrow \text{loc} \\
\text{org} & \rightarrow \text{org\_term}\,^+ \\
\text{org\_term} & \rightarrow \text{Brown} \\
\text{org\_term} & \rightarrow \text{University} \\
\text{pers} & \rightarrow \text{pers\_term}\,^+ \\
\text{pers\_term} & \rightarrow \text{Moses} \\
\text{pers\_term} & \rightarrow \text{Brown}
\end{align*}\]
Internal structure

Grammar:

\[ \text{NE} \rightarrow \text{org} \]
\[ \text{org} \rightarrow \text{org}^1 \text{org}^2 \]

\[ \text{org}^1 \rightarrow \text{Brown} \]
\[ \text{org}^2 \rightarrow \text{University} \]
Internal structure

Grammar:

\[ NE \rightarrow org \]
\[ org \rightarrow org^1 \ org^2 \]
\[ org \rightarrow (org^1)(org^2)(org^3)(org^4)(org^5) \]
\[ org^1 \rightarrow \text{Brown} \]
\[ org^2 \rightarrow \text{University} \]
Multiword expansions

Grammar:

\[ NE \rightarrow \text{loc} \]
\[ \text{place} \rightarrow \text{loc}^1 \text{loc}^2 \]
\[ \text{loc}^1 \rightarrow \text{Providence} \]
\[ \text{loc}^2 \rightarrow \text{Rhode Island} \]
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:

\[
\begin{align*}
NE & \rightarrow \text{org pronouns}_{\text{org}} \\
\text{org} & \rightarrow \text{org}^1 \text{org}^2 \\
\text{pronouns}_{\text{org}} & \rightarrow \# \text{pronoun}_{\text{org}}^* \\
\text{pronoun}_{\text{org}} & \rightarrow \text{which} \\
\text{pronoun}_{\text{org}} & \rightarrow \text{they} \\
\ldots \\
\text{pronoun}_{\text{org}} & \rightarrow \text{he} \\
\ldots
\end{align*}
\]

\[
\begin{align*}
\text{NE} & \quad \text{org} \\
\text{org}^1 & \quad \text{Brown} \\
\text{org}^2 & \quad \text{University} \\
\# & \quad \text{which} \\
\text{pronouns}_{\text{org}} & \\
\end{align*}
\]
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.

<table>
<thead>
<tr>
<th>Ordered template</th>
<th>pers¹</th>
<th>pers²</th>
<th>pers³</th>
<th>pers⁴</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof.</td>
<td>Mark</td>
<td>E.</td>
<td>Johnson</td>
<td></td>
</tr>
</tbody>
</table>
Modeling consistency

Model’s concept of consistency follows (Charniak ‘01):

Phrases are consistent if none of their internal subparts clash.

Ordered template

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.

<table>
<thead>
<tr>
<th>Ordered template</th>
<th>pers\textsuperscript{1}</th>
<th>pers\textsuperscript{2}</th>
<th>pers\textsuperscript{3}</th>
<th>pers\textsuperscript{4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Mark E. Johnson</td>
<td>Prof. Mark E. Johnson</td>
<td>Prof. Mark E. Johnson</td>
<td>Prof. Mark E. Johnson</td>
<td>Prof. Mark E. Johnson</td>
</tr>
</tbody>
</table>

realizations
Modeling consistency

Model’s concept of consistency follows (Charniak ‘01):

Phrases are consistent if none of their internal subparts clash.

Ordered template

<table>
<thead>
<tr>
<th>pers^1</th>
<th>pers^2</th>
<th>pers^3</th>
<th>pers^4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof.</td>
<td>Mark</td>
<td>E.</td>
<td>Johnson</td>
</tr>
</tbody>
</table>

realizations

<table>
<thead>
<tr>
<th>pers^1</th>
<th>pers^2</th>
<th>pers^3</th>
<th>pers^4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof.</td>
<td>Mark</td>
<td>Johnson</td>
<td>Johnson</td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Modeling consistency

Model’s concept of consistency follows (Charniak ‘01):

Phrases are consistent if none of their internal subparts clash.

<table>
<thead>
<tr>
<th>Ordered template</th>
<th>pers(^1)</th>
<th>pers(^2)</th>
<th>pers(^3)</th>
<th>pers(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof.</td>
<td>Mark</td>
<td>E.</td>
<td>Johnson</td>
<td></td>
</tr>
</tbody>
</table>

realizations

<table>
<thead>
<tr>
<th></th>
<th>pers(^1)</th>
<th>pers(^2)</th>
<th>pers(^3)</th>
<th>pers(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof.</td>
<td>Mark</td>
<td>Johnson</td>
<td>Johnson</td>
<td></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

inconsistent

<table>
<thead>
<tr>
<th>pers(^1)</th>
<th>pers(^2)</th>
<th>pers(^3)</th>
<th>pers(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark</td>
<td>Steedman</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview

Introduction

Clustering as parsing

Consistency: finding possible entities

Experiments: pronouns are key!

Future directions
Datasets:

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

Combine features for multiple examples:

<table>
<thead>
<tr>
<th>Hillary Clinton #</th>
<th>#</th>
<th>#</th>
<th>who</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hillary Clinton #</td>
<td>Secretary #</td>
<td>#</td>
<td>she</td>
</tr>
<tr>
<td>Hillary Clinton #</td>
<td>#</td>
<td>spokesman-for #</td>
<td>her</td>
</tr>
</tbody>
</table>

More data in equal time... but no per-document features.
Basic results

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

Confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>loc</th>
<th>org</th>
<th>per</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>1187</td>
<td>97</td>
<td>37</td>
</tr>
<tr>
<td>ORG</td>
<td>223</td>
<td>1517</td>
<td>122</td>
</tr>
<tr>
<td>PER</td>
<td>36</td>
<td>20</td>
<td>820</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (All ORG)</td>
<td>42.5</td>
</tr>
<tr>
<td>Core NPs (no consistency)</td>
<td>45.5</td>
</tr>
<tr>
<td>Core NPs (consistency)</td>
<td>48.5</td>
</tr>
<tr>
<td>Context features (nominal/prep)</td>
<td>83.3</td>
</tr>
<tr>
<td>All features (context + pronouns)</td>
<td>87.1</td>
</tr>
</tbody>
</table>
## Named entity structure

<table>
<thead>
<tr>
<th>pers&lt;sup&gt;0&lt;/sup&gt;</th>
<th>pers&lt;sup&gt;1&lt;/sup&gt;</th>
<th>pers&lt;sup&gt;2&lt;/sup&gt;</th>
<th>pers&lt;sup&gt;3&lt;/sup&gt;</th>
<th>pers&lt;sup&gt;4&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>rep.</td>
<td>john</td>
<td>minister</td>
<td>brown</td>
<td>jr.</td>
</tr>
<tr>
<td>sen.</td>
<td>robert</td>
<td>j.</td>
<td>smith</td>
<td>a</td>
</tr>
<tr>
<td>washington</td>
<td>david</td>
<td>john</td>
<td>b</td>
<td>smith</td>
</tr>
<tr>
<td>dr.</td>
<td>michael</td>
<td>l.</td>
<td>johnson</td>
<td>iii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>loc&lt;sup&gt;0&lt;/sup&gt;</th>
<th>loc&lt;sup&gt;1&lt;/sup&gt;</th>
<th>loc&lt;sup&gt;2&lt;/sup&gt;</th>
<th>loc&lt;sup&gt;3&lt;/sup&gt;</th>
<th>loc&lt;sup&gt;4&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>washington</td>
<td>the</td>
<td>texas</td>
<td>county</td>
<td>monday</td>
</tr>
<tr>
<td>los angeles</td>
<td>st.</td>
<td>new york</td>
<td>city</td>
<td>thursday</td>
</tr>
<tr>
<td>south</td>
<td>new</td>
<td>washington</td>
<td>beach</td>
<td>river</td>
</tr>
<tr>
<td>north</td>
<td>national</td>
<td>united states</td>
<td>valley</td>
<td>tuesday</td>
</tr>
</tbody>
</table>
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 nonparametereic Bayesian inference experiments on unsupervised word segmentation with adaptor grammars”, Mark Johnson and Sharon Goldwater.
Adaptor grammars (Johnson+al ‘07)

Prior grammar:

\[
\begin{align*}
\text{count rule} \\
1 & \quad \text{words} \rightarrow \text{word words} \\
1 & \quad \text{words} \rightarrow \text{word} \\
1 & \quad \text{word} \rightarrow \text{Rhode} \\
1 & \quad \text{word} \rightarrow \text{Island} \\
1 & \quad \text{word} \rightarrow \text{Colorado} \\
\ldots \\
1 & \quad \text{loc}^2 \rightarrow \text{words}
\end{align*}
\]

Data:

- Providence Rhode Island
- Boulder Colorado
- Newport Rhode Island
- ...
Adaptor grammars (Johnson+al ’07)

Posterior grammar:

- **count rule**
  - 2 words → word words
  - 2 words → word
  - 2 word → Rhode
  - 2 word → Island
  - 1 word → Colorado
  - ...
  - 1 \( loc^2 \) → words
  - 1 \( loc^2 \) → Rhode Island

Data:

```
    NE
     / \   \
loc 1  loc 2
     /    /   \
loc words words
     |     |    \\
word word word
     \\
    Providence Rhode Island
    \\
Boulder Colorado
    \\
Newport Rhode Island
```
Adaptor grammars (Johnson+al ‘07)

Posterior grammar:

**count**  **rule**

2  **words** →  **word**  **words**
3  **words** →  **word**
2  **word** →  **Rhode**
2  **word** →  **Island**
2  **word** →  **Colorado**

...  

1  **loc** →  **words**
1  **loc** →  **Rhode Island**
1  **loc** →  **Colorado**

Data:

```
NE
   ↓
loc
   ↓
loc 1
   ↓
word
   ↓
words
   ↓
Providence
   ↓
loc 1
   ↓
word
   ↓
words
   ↓
Rhode Island
   ↓
loc 1
   ↓
word
   ↓
words
   ↓
word
   ↓
Colorado
   ↓
loc 2
   ↓
word
   ↓
words
   ↓
Newport
   ↓
Rhode Island
```
Adaptor grammars (Johnson+al ‘07)

**Posterior grammar:**

*count rule*

2 *words* → *word* *words*

3 *words* → *word*

2 *word* → *Rhode*

2 *word* → *Island*

2 *word* → *Colorado*

...  

1 \underline{loc^2} → *words*

2 \underline{loc^2} → *Rhode Island*

1 \underline{loc^2} → *Colorado*

Data:
Overview

Adaptor grammars: framework for Bayesian grammar learning

Implementing Consistency

Inference: a general problem for this approach
Implementing consistency

Grammar:

\[ NE \rightarrow \text{org} \]
\[ \text{org} \rightarrow \text{org}_{\text{Brown}} \ldots \]
\[ \text{org}_{\text{Brown}} \rightarrow \text{org}^1_{\text{Brown}} \text{org}^2_{\text{Brown}} \]
\[ \text{org}^1_{\text{Brown}} \rightarrow \text{org}^1 \]
\[ \text{org}^2_{\text{Brown}} \rightarrow \text{org}^2 \]
\[ \text{org}^1 \rightarrow \text{Brown} \]
\[ \text{org}^2 \rightarrow \text{University} \]

Underlined nonterminals are Dirichlet processes. \(\text{org}^1_{\text{Brown}}\) and \(\text{org}^2_{\text{Brown}}\) 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 $\textit{loc}^2$:

1. $\textit{loc}^2 \rightarrow \textit{words}$
2. $\textit{loc}^2 \rightarrow \text{Colorado}$
3. $\textit{loc}^2 \rightarrow \text{Rhode Island}$

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 $loc^2$:

1. $loc^2 ightarrow words$
2. $loc^2 ightarrow Colorado$
3. $loc^2 ightarrow Rhode Island$

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 $\text{loc}^2$:

1. $\text{loc}^2 \rightarrow \text{words}$
2. $\text{loc}^2 \rightarrow \text{Colorado}$
3. $\text{loc}^2 \rightarrow \text{Rhode Island}$

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 \( \text{loc}^2 \):

1. \( \text{loc}^2 \rightarrow \text{words} \)
2. \( \text{loc}^2 \rightarrow \text{Colorado} \)
3. \( \text{loc}^2 \rightarrow \text{Rhode Island} \)
4. \( \text{loc}^2 \rightarrow \text{Rhode} \)

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 $loc^2$:

1. $loc^2 \rightarrow \text{words}$
2. $loc^2 \rightarrow \text{Colorado}$
3. $loc^2 \rightarrow \text{Rhode Island}$
4. $loc^2 \rightarrow \text{Rhode}$

Data:

- Providence
- Rhode Island
- Newport
- Rhode Island
- Boulder
- Colorado
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 $loc^2$:

1. $loc^2 \rightarrow \text{words}$
2. $loc^2 \rightarrow \text{Colorado}$
3. $loc^2 \rightarrow \text{Rhode}$
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 $\text{loc}^2$:

1. $\text{loc}^2 \rightarrow \text{words}$
2. $\text{loc}^2 \rightarrow \text{Colorado}$
3. $\text{loc}^2 \rightarrow \text{Rhode}$

Data:

- Providence Rhode Island
- Boulder Colorado
- Newport Rhode Island
- Rhode Island
- Rhode Island
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