Lexical Category Acquisition as an Incremental Process

Afra Alishahi, Grzegorz Chrupała

FEAST, July 21, 2009
Children’s Sensitivity to Lexical Categories

- Gelman & Taylor’84: 2-year-olds treat names not followed by a determiner (e.g. “Zav”) as a proper name, and interpret them as individuals (e.g., the animal-like toy).
Children’s Sensitivity to Lexical Categories

- Gelman & Taylor’84: 2-year-olds treat names followed by a determiner (e.g. “the zav”) as a common name, and interpret them as category members (e.g., the block-like toy).
Challenges of Learning Lexical Categories

- Children form lexical categories gradually and over time
  - Nouns and verb categories are learned by age two, but adjectives are not learned until age six
- Child language acquisition is bounded by memory and processing limitations
  - Child category learning is unsupervised and incremental
  - Highly extensive processing of data is cognitively implausible
- Natural language categories are not clear cut
  - Many words are ambiguous and belong to more than one category
  - Many words appear in the input very rarely
Goals

• Propose a cognitively plausible algorithm for inducing categories from child-directed speech

• Suggest a novel way of evaluating the learned categories via a variety of language tasks
Part I: Category Induction
Information Sources

- Children might use different information cues for learning lexical categories
  - perceptual cues (phonological and morphological features)
  - semantic properties of the words
  - distributional properties of the local context each word appears in

- Distributional context is a reliable cue
  - Analysis of child-directed speech shows abundance of consistent contextual patterns (Redington et al., 1998; Mintz, 2003)
  - Several computational models have used distributional context to induce intuitive lexical categories (e.g. Schutze 1993, Clark 2000)
Computational Models of Lexical Category Induction

• Hierarchical clustering models
  • Starting from a cluster per each word type, the two most similar clusters are merged in each iteration (Schutze’93, Redington et al’98)

• Cluster optimization models
  • Vocabulary is partitioned into non-overlapping clusters, which are optimized according to an information theoretic measure (Brown’92, Clark’00)

• Incremental clustering models
  • Each word usage is added to the most similar existing cluster, or a new cluster is created (e.g. Cartwright & Brent’97, Parisien et al’08)

• Existing models rely on optimizing techniques, demanding high computational load for processing data
Our Model

• We propose an efficient incremental model for lexical category induction from unannotated text

• Word usages are categorized based on similarity of their content and context to the existing categories

  “want to put them on”

• Each usage is represented as a vector:

<table>
<thead>
<tr>
<th>-2=want</th>
<th>-1=to</th>
<th>0=put</th>
<th>1=them</th>
<th>2=on</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Representation of Word Categories

- A lexical category is a cluster of word usages
  - The distributional context of a category is represented as the mean of the distribution vectors of its members

<table>
<thead>
<tr>
<th>-2=want</th>
<th>-2=have</th>
<th>-1=to</th>
<th>0=go</th>
<th>0=sit</th>
<th>0=show</th>
<th>0=send</th>
<th>1=it</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.75</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>...</td>
</tr>
</tbody>
</table>

- The similarity between two clusters is measured by the dot product of their vectors
Online Clustering Algorithm

For every word usage $w$:

- Create new cluster $C_{\text{new}}$
- Add $\Phi(w)$ to $C_{\text{new}}$
- $C_w = \arg\max_{C \in \text{Clusters}} \text{Similarity}(C_{\text{new}}, C)$
- If $\text{Similarity}(C_{\text{new}}, C_w) \geq \theta_w$
  - merge $C_w$ and $C_{\text{new}}$
  - $C_{\text{next}} = \arg\max_{C \in \text{Clusters} - \{C_w\}} \text{Similarity}(C_w, C)$
  - If $\text{Similarity}(C_w, C_{\text{next}}) \geq \theta_c$
    * merge $C_w$ and $C_{\text{next}}$

where $\text{Similarity}(x, y) = x \cdot y$ and the vector $\Phi(w)$ represents the context features of the current word usage $w$. 

Experimental Data

• Manchester corpus from CHILDES database (Theakston et al.’01, MacWhinney’00)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Corpus</th>
<th>#Sentences</th>
<th>#Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop</td>
<td>Anne</td>
<td>857</td>
<td>3,318</td>
</tr>
<tr>
<td>Train</td>
<td>Anne</td>
<td>13,772</td>
<td>73,032</td>
</tr>
<tr>
<td>Test</td>
<td>Becky</td>
<td>1,116</td>
<td>5,431</td>
</tr>
</tbody>
</table>

(One-word sentences are excluded from training and test data)

• Threshold values are set based on development data:
  \[ \theta_w = 2^7 \times 10^{-3} \quad \text{and} \quad \theta_c = 2^{10} \times 10^{-3} \]
Category Size

Distribution of the size of categories

Coverage of tokens by categories

Processing the training data yielded a total of 427 categories.
Sample Induced Categories

<table>
<thead>
<tr>
<th>Most frequent values for the <em>content word</em> feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>do</td>
</tr>
<tr>
<td>are</td>
</tr>
<tr>
<td>will</td>
</tr>
<tr>
<td>have</td>
</tr>
<tr>
<td>can</td>
</tr>
<tr>
<td>has</td>
</tr>
<tr>
<td>does</td>
</tr>
<tr>
<td>had</td>
</tr>
<tr>
<td>were</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Most frequent values for the <em>previous word</em> feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘s</td>
</tr>
<tr>
<td>is</td>
</tr>
<tr>
<td>was</td>
</tr>
<tr>
<td>in</td>
</tr>
<tr>
<td>then</td>
</tr>
<tr>
<td>goes</td>
</tr>
<tr>
<td>on</td>
</tr>
</tbody>
</table>

| bit  |
| little |
| good |
| big  |
| very |
| long |
| few  |
| drink |
| funny |

| ‘re  |
| ‘ve |
| want |
| got  |
| see  |
| were |
| were |
| do   |
| find |
| going |

Most frequent values for the *content word* feature

Most frequent values for the *previous word* feature
• The growth of the size of the vocabulary (i.e. word types), as well as the number of lexical categories, slows down over time
Part 2: Evaluation
Common Evaluation Approach

- POS tags as gold-standard: evaluate their categories based on how well they match POS categories
  - Accuracy and Recall: every pair or words in an induced category should belong to the same POS category (Redington et al.’98)
  - Order of category formation: categories that resemble POS categories show the same developmental trend (Parisien et al.’08)

- Alternative evaluation techniques
  - Substitutability of category members in training sentences (Frank et al.’09)
  - Perplexity of a finite state model based on two sets of categories (Clark’01)
Our Proposal: Measuring ‘Usefulness’ instead of ‘Correctness’

• Instead of using a gold-standard to compare our categories against, we use the categories in a variety of applications
  • Word prediction from context
  • Inferring semantic properties of novel words based on the context they appear in
• We compare the performance in each task against a POS-based implementation of the same task
Word Prediction

- Task: predicting a missing (target) word based on its context
  - This task is non-deterministic (i.e. it can have many answers), but the context can significantly limit the choices
- Human subjects have shown to be remarkably accurate at using context for guessing target words (Gleitman’90, Lesher’02)

She slowly --- the road
I had --- for lunch
Word Prediction - Methodology

Test item:

<table>
<thead>
<tr>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>want</td>
<td>to</td>
<td>put</td>
<td>them</td>
<td>on</td>
</tr>
</tbody>
</table>
Word Prediction - Methodology

Test item:

\[
\begin{array}{c|c|c}
-2 & -1 & 0 \\
\hline
want & to & put \\
\hline
1 & 2 & item \quad on
\end{array}
\]
Word Prediction - Methodology

Test item:

<table>
<thead>
<tr>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>want</td>
<td>to</td>
<td>put</td>
<td>them</td>
<td>on</td>
</tr>
</tbody>
</table>

Categorize $C_w$

<table>
<thead>
<tr>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Word Prediction - Methodology

Test item:

```
-2 -1  0  1  2
want to  put  them on
```

Categorize

\( C_w \)

Ranked word list for content feature

```
-2 -1  0  1  2
... ... ...
```

```
make
take
get
put
sit
eat
let
point
give
```

```
Word Prediction - Methodology

Test item:

-2 -1 0 1 2
want to put them on

Categorize

\( C_w \)

Ranked word list for content feature

Reciprocal rank of the target word: 1/4

make
take
give
get
put
sit
eat
let
point

... ...
...
...
...

put
Word Prediction - POS Categories

baby 's Mummy
n v n:prop

put them on the table look
v pro prep det n v

have her hair brushed
v pro n part

there is a spider
adv:loc v det n

...
Word Prediction - POS Categories

Labelled Data

- baby's Mummy
  n v n:prop

- put them on the table
  v pro prep det n v

- have her hair brushed
  v pro n part

- there is a spider
  adv:loc v det n

Noun Category

- baby
- table
- hair
- spider
- ...

...
Word Prediction - POS Categories

Labelled Data

Noun Category

Feature Representation

baby's Mummy
n v n:prop
put them on the table look v pro prep det n v
have her hair brushed v pro n part
there is a spider adv:loc v det n

...
# Word Prediction - Results

<table>
<thead>
<tr>
<th>Category Type</th>
<th>Mean Reciprocal Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>0.073</td>
</tr>
<tr>
<td>Induced</td>
<td>0.198</td>
</tr>
<tr>
<td>Word type</td>
<td>0.009</td>
</tr>
</tbody>
</table>
Inferring Word Semantic Properties

- Task: guessing the semantic properties of a novel word based on its local context

- Children and adults can guess (some aspects of) the meaning of a novel word from context (Landau & Gleitman’85, Naigles & Hoff-Ginsberg’95)

I had ZAV for lunch
Word Semantic Properties

• Semantic features of each word are extracted from WordNet:

  - **Semantic vector for cake**
  - WordNet hypernyms for cake
    - cake
      - baked goods
      - food
      - solid
      - substance, matter

• Semantic feature vector for each category is the mean of the semantic vectors of its members

• Note: semantic features are not used in categorization
Word Semantic Properties

• Semantic features of each word are extracted from WordNet:

- Semantic feature vector for each category is the mean of the semantic vectors of its members

• Semantic feature vector for each category is the mean of the semantic vectors of its members

• Note: semantic features are not used in categorization
Inferring Semantic Properties - Methodology

Test item:

<table>
<thead>
<tr>
<th></th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>ate</td>
<td>Zag</td>
<td></td>
<td>for</td>
<td>lunch</td>
</tr>
</tbody>
</table>
Inferring Semantic Properties - Methodology

Test item:

<table>
<thead>
<tr>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>ate</td>
<td>Zag</td>
<td>for</td>
<td>lunch</td>
</tr>
</tbody>
</table>
Inferring Semantic Properties - Methodology

Test item:

<table>
<thead>
<tr>
<th></th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I ate Zag for lunch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Categorize

$C_w$

<table>
<thead>
<tr>
<th></th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Inferring Semantic Properties - Methodology

Test item:

Categorize

Semantic feature for target word position

entity
object
substance
matter
food
edible:
Inferring Semantic Properties - Methodology

Test item:

\[
\begin{array}{cccc}
-2 & -1 & 0 & 1 & 2 \\
I & ate & Zag & for & lunch \\
\end{array}
\]

Categorize

\[C_w\]

Semantic feature for target word position

original target word:

\[
0 \quad soup
\]

entity
object
substance
matter
food
edible

: 
Inferring Semantic Properties - Methodology

Test item:

<table>
<thead>
<tr>
<th></th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ate</td>
<td>Zag</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>for</td>
<td>lunch</td>
</tr>
</tbody>
</table>

Categorize

\[ C_w \]

Semantic feature for target word position

original target word:

\[ 0 \]

soup

Semantic vector

entity
object
substance
matter
food
edible
liquid
meal
soup

:
Inferring Semantic Properties - Methodology

Test item:

<table>
<thead>
<tr>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ate</td>
<td>Zag for lunch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Categorize

Semantic feature for target word position

original target word:

0

soup

Semantic vector

entity
object
substance
matter
food
edible
liquid
meal
soup

Semantic feature for target word position

Similarity Measure
### Inferring Semantic Properties - Results

<table>
<thead>
<tr>
<th>Category Type</th>
<th>Average Dot Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>0.035</td>
</tr>
<tr>
<td>Induced</td>
<td>0.048</td>
</tr>
</tbody>
</table>
Discussion

• We propose an incremental model of lexical category acquisition based on distributional properties of words.
  
  • Model learns intuitive categories from child-directed speech.
  
  • Categories are successfully used in word prediction and the inference of semantic properties of words from context.

• Finer-grained lexical categories seem more suitable for some tasks than traditional POS categories.
  
  • Standardized applications are needed to evaluate and compare lexical categories induced by different unsupervised methods.
Future Directions

• Improving the model
  • Alternative representations of the local context
    • Applying a Gaussian filter on context window
  • Bootstrapping
    • Using categories of the previous words as feature
  • Alternative representations of categories and similarity measures

• Evaluating categories via more applications
  • Lexical decision
  • Grammaticality judgment