Toward a Psycholinguistically-Motivated Model of Language Processing

William Schuler\textsuperscript{1},
Samir AbdelRahman\textsuperscript{2},
Tim Miller\textsuperscript{1},
Lane Schwartz\textsuperscript{1}

June 24, 2011

\textsuperscript{1}University of Minnesota
\textsuperscript{2}Cairo University
Background

NSF project: implement interactive model of speech/language processing

- Parsing/speech recognition dep. on semantic interpretation in context
  (Tanenhaus et al., 1995, 2002)
NSF project: implement **interactive model** of speech/language processing

- Parsing/speech recognition dep. on semantic interpretation in context (Tanenhaus et al., 1995, 2002)
- Factored time-series model of speech recognition, parsing, interpretation (formal model presented in Computational Linguistics, in press)
- Real-time interactive speech interface: define new objects, then refer (implemented system presented at IUI’08; interp. → vectors of objects)
- This year: interp. vector → head word probabilities / LSA semantics
NSF project: implement **interactive model** of speech/language processing

- Parsing/speech recognition dep. on semantic interpretation in context (Tanenhaus et al., 1995, 2002)
- Factored time-series model of speech recognition, parsing, interpretation (formal model presented in Computational Linguistics, in press)
- Real-time interactive speech interface: define new objects, then refer (implemented system presented at IUI’08; interp. → vectors of objects)
- This year: interp. vector → head word probabilities / LSA semantics
- **Why time-series?** composition expensive; time-series simpler than CKY
Background

NSF project: implement interactive model of speech/language processing

- Parsing/speech recognition dep. on semantic interpretation in context (Tanenhaus et al., 1995, 2002)
- Factored time-series model of speech recognition, parsing, interpretation (formal model presented in Computational Linguistics, in press)
- Real-time interactive speech interface: define new objects, then refer (implemented system presented at IUI’08; interp. $\rightarrow$ vectors of objects)
- This year: interp. vector $\rightarrow$ head word probabilities / LSA semantics
- Why time-series? composition expensive; time-series simpler than CKY
- Today: is it safe? Human-like memory limits still parse most sentences (evaluated on broad-coverage WSJ Treebank)
Background

NSF project: implement interactive model of speech/language processing

- Parsing/speech recognition dep. on semantic interpretation in context (Tanenhaus et al., 1995, 2002)
- Factored time-series model of speech recognition, parsing, interpretation (formal model presented in Computational Linguistics, in press)
- Real-time interactive speech interface: define new objects, then refer (implemented system presented at IUI’08; interp. → vectors of objects)
- This year: interp. vector → head word probabilities / LSA semantics
- Why time-series? composition expensive; time-series simpler than CKY
- Today: is it safe? Human-like memory limits still parse most sentences (evaluated on broad-coverage WSJ Treebank)
- Friday: model transform also gives nice explanation of speech repair (evaluated on Switchboard Treebank)
Early work:
Marcus ('80), Abney & Johnson ('91), Gibson ('91), Lewis ('93), ... —
Garden pathing, processing difficulties due to memory saturation

- processing difficulties also due to other factors, e.g. similarity (Miller & Chomsky '63; Lewis '93), decay (Gibson '98)
- favor left-corner; but eager/deferred composition? → parallel proc.
Early work:
Marcus ('80), Abney & Johnson ('91), Gibson ('91), Lewis ('93), ...

Garden pathing, processing difficulties due to memory saturation

- processing difficulties also due to other factors, e.g. similarity (Miller & Chomsky '63; Lewis '93), decay (Gibson '98)
- favor left-corner; but eager/deferred composition? → parallel proc.

More recently:
Hale (2003), Levy (2008)

Difficulties due to changing probability/activation of competing hypotheses

- empirical success
- decouples processing difficulty from memory saturation
- but does not explain how/whether parsing fits in short-term memory (and parsing should now be comfortably within STM, not at limit!)
This model:
Explicit memory elements, compatible w. interactive interpretation

- Bounded store of incomplete referents, constituents over time
  - incomplete referents: individual/group of objects/events (∼ Haddock’89)
  - incomplete constituents: e.g. S/NP (S w/o NP; ∼ CCG, Steedman’01)
This model:
Explicit memory elements, compatible w. interactive interpretation

▶ Bounded store of incomplete referents, constituents over time
  ▶ incomplete referents: individual/group of objects/events (∼ Haddock’89)
  ▶ incomplete constituents: e.g. S/NP (S w/o NP; ∼ CCG, Steedman’01)

▶ For simplicity, strict complexity limit on memory elements (no chunks):
  one incomplete referent/constituent per memory element
This model:
Explicit memory elements, compatible w. interactive interpretation

- Bounded store of incomplete referents, constituents over time
  - incomplete referents: individual/group of objects/events (≈ Haddock’89)
  - incomplete constituents: e.g. S/NP (S w/o NP; ≈ CCG, Steedman’01)

- For simplicity, strict complexity limit on memory elements (no chunks):
  one incomplete referent/constituent per memory element

- Sequence of stores ⇔ phrase structure via simple tree transform
  (≈Johnson’98; system ≈Roark’01/Henderson’04 but mem-optimized)
This model:
Explicit memory elements, compatible w. interactive interpretation

- Bounded store of incomplete referents, constituents over time
  - incomplete referents: individual/group of objects/events (∼ Haddock’89)
  - incomplete constituents: e.g. S/NP (S w/o NP; ∼ CCG, Steedman’01)
- For simplicity, strict complexity limit on memory elements (no chunks):
  one incomplete referent/constituent per memory element
- Sequence of stores ⇔ phrase structure via simple tree transform
  (∼Johnson’98; system ∼Roark’01/Henderson’04 but mem-optimized)
- Alternative stores active in pockets, not monolithic (unbounded beam)
- Essentially, factored HMM-like time-series model
This model:
Explicit memory elements, compatible w. interactive interpretation

- Bounded store of incomplete referents, constituents over time
  - incomplete referents: individual/group of objects/events (∼ Haddock’89)
  - incomplete constituents: e.g. S/NP (S w/o NP; ∼ CCG, Steedman’01)
- For simplicity, strict complexity limit on memory elements (no chunks):
  one incomplete referent/constituent per memory element
- Sequence of stores ⇔ phrase structure via simple tree transform
  (∼Johnson’98; system ∼Roark’01/Henderson’04 but mem-optimized)
- Alternative stores active in pockets, not monolithic (unbounded beam)
- Essentially, factored HMM-like time-series model

Evaluation of Coverage:
- Can parse nearly 99.96% of WSJ 2–21 using ≤ 4 memory elements
Hierarchic Hidden Markov Model

Factored HMM model (Murphy & Paskin ’01): bounded probabilistic PDA

\[
\hat{h}^{1..D}_{1..T} \overset{\text{def}}{=} \arg\max_{h^{1..D}_{1..T}} \prod_{t=1}^{T} P_{\Theta_{LM}}(h_t^{1..D} | h_{t-1}^{1..D}) \cdot P_{\Theta_{OM}}(o_t | h_t^{1..D})
\]
Hierarchic Hidden Markov Model

Factored HMM model (Murphy & Paskin ’01): bounded probabilistic PDA

\[
P_{\text{LM}}(q_{t:D} \mid q_{t-1:D}) = \sum_{f_{t:D}} \prod_{d=1}^{D} P_{\rho}(f_t^d \mid f_t^{d+1} q_t^d q_{t-1}^d q_{t-1}^{d-1}) \cdot P_{\sigma}(q_t^d \mid f_t^{d+1} f_t^d q_{t-1}^d q_{t-1}^{d-1})
\]

def \(=\) 

Schuler, AbdelRahman, Miller, Schwartz
Psycholinguistically-Motivated Model of Language Processing
Derive model probabilities from training trees:

Must be transformed into flat, memory-efficient form...
‘Right-corner transform’: $\sim$ left-corner, but reversed so incomplete on right.
Align levels to a grid, to train HHMM:

\[
\begin{array}{cccccccccccccc}
  & t=1 & t=2 & t=3 & t=4 & t=5 & t=6 & t=7 & t=8 & t=9 & t=10 & t=11 & t=12 & t=13 & t=14 & t=15 \\
\hline
\text{d=1} & \text{NP} & \text{NN} & \text{NP} & \text{PP} & \text{NP} & \text{NP} & \text{NP} & \text{NP} & \text{NP} & \text{NP} & \text{NNS} & \text{NNS} & \text{S} & \text{NP} & \text{S} \\
\text{d=2} & & & & & & & & & & & & & & & \\
\text{d=3} & & & & & & & & & & & & & & & \\
\text{word} & \text{strong} & \text{demand} & \text{for} & \text{new} & \text{york} & \text{city} & \text{general} & \text{obligation} & \text{bonds} & \text{proped} & \text{up} & \text{the} & \text{municipal} & \text{market} & \\
\end{array}
\]
Mapping to HHMM

Align levels to a grid, to train HHMM:

Different than other left-corner models: not all levels open for adjunction. Many configs in parallel; weights depend on learned HHMM probabilities.
Tree Transform

Transform is very simple — first flatten out right-recursive structure:

\[
\begin{align*}
A_1 & \quad \alpha_1 \\
\alpha_2 & \quad A_2 \\
\alpha_3 & \quad A_3 \\
\Rightarrow & \quad A_1 / A_2 \quad A_2 / A_3 \quad A_3 \\
& \quad \alpha_1 \\
& \quad \alpha_2 \\
& \quad \alpha_3 \\
A_1 & \quad \alpha_1 \\
\alpha_2 & \quad A_2 \\
\alpha_3 & \quad A_3 \\
\Rightarrow & \quad A_1 / A_2 \quad A_2 / A_3 \\
& \quad \alpha_1 \\
& \quad \alpha_2 \\
& \quad \alpha_3 \\
& \ldots
\end{align*}
\]

then replace it with left-recursive structure:

\[
\begin{align*}
A_1 & \quad \alpha_1 \\
\Rightarrow & \quad A_1 / A_2 \quad A_2 / A_3 \quad \alpha_3 \\
& \quad \alpha_2 \\
& \quad A_1 / A_2 \quad A_2 / A_3 \quad \alpha_3 \\
& \quad \alpha_2 \\
& \ldots
\end{align*}
\]
Tree Transform

Transform is very simple — first flatten out right-recursive structure:

\[
\begin{align*}
A_1 & \quad A_2 & \quad A_3 \\
\alpha_1 & \quad \alpha_2 & \quad \alpha_3
\end{align*}
\]

\[ \Rightarrow \quad A_1/A_2 \quad A_2/A_3 \quad A_3, \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \]

then replace it with left-recursive structure:

\[
\begin{align*}
A_1 & \quad A_2 & \quad A_3 \\
\alpha_1 & \quad \alpha_2 & \quad \alpha_3
\end{align*}
\]

\[ \Rightarrow \quad A_1/A_2 \quad A_2/A_3 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \]

Only right recursion remaining is center embedding, known to be limited:

“The cart the horse the man bought pulled broke.”

(Miller and Chomsky, 1963)
Coverage

How many levels do you need? About four.

<table>
<thead>
<tr>
<th>stack memory capacity</th>
<th>sentences</th>
<th>coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>no stack memory</td>
<td>127</td>
<td>0.32%</td>
</tr>
<tr>
<td>1 stack element</td>
<td>3,496</td>
<td>8.78%</td>
</tr>
<tr>
<td>2 stack elements</td>
<td>25,909</td>
<td>65.05%</td>
</tr>
<tr>
<td>3 stack elements</td>
<td>38,902</td>
<td>97.67%</td>
</tr>
<tr>
<td><strong>4 stack elements</strong></td>
<td><strong>39,816</strong></td>
<td><strong>99.96%</strong></td>
</tr>
<tr>
<td>5 stack elements</td>
<td>39,832</td>
<td>100.00%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>39,832</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Percent coverage of transformed treebank sections 2–21 w. no punctuation

Good! Because that’s supposed to be our limit! (Cowan, 2001)
Coverage

How many levels do you need? About four.

<table>
<thead>
<tr>
<th>stack memory capacity</th>
<th>sentences</th>
<th>coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>no stack memory</td>
<td>127</td>
<td>0.32%</td>
</tr>
<tr>
<td>1 stack element</td>
<td>3,496</td>
<td>8.78%</td>
</tr>
<tr>
<td>2 stack elements</td>
<td>25,909</td>
<td>65.05%</td>
</tr>
<tr>
<td>3 stack elements</td>
<td>38,902</td>
<td>97.67%</td>
</tr>
<tr>
<td>4 stack elements</td>
<td>39,816</td>
<td>99.96%</td>
</tr>
<tr>
<td>5 stack elements</td>
<td>39,832</td>
<td>100.00%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>39,832</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Percent coverage of transformed treebank sections 2–21 w. no punctuation

Good! Because that’s supposed to be our limit! (Cowan, 2001)

Now, a windfall in accuracy due to pruned search space?
No... guessing open adjunction sites to save memory holds back accuracy

Accuracy results w. no lexicalization or smoothing:

<table>
<thead>
<tr>
<th></th>
<th>LP</th>
<th>LR</th>
<th>F</th>
<th>fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>with punctuation:  (≤ 40 wds)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KM’03: unmodified, devset</td>
<td>–</td>
<td>–</td>
<td>72.6</td>
<td>0</td>
</tr>
<tr>
<td>KM’03: par+sib, devset</td>
<td>–</td>
<td>–</td>
<td>77.4</td>
<td>0</td>
</tr>
<tr>
<td>CKY: binarized, devset</td>
<td>72.3</td>
<td>71.1</td>
<td>71.7</td>
<td>0</td>
</tr>
<tr>
<td><strong>HHMM: par+sib, devset</strong></td>
<td><strong>81.4</strong></td>
<td><strong>82.9</strong></td>
<td><strong>82.1</strong></td>
<td><strong>1.4</strong></td>
</tr>
<tr>
<td>CKY: binarized, sect 23</td>
<td>72.0</td>
<td>69.7</td>
<td>70.8</td>
<td>0.3</td>
</tr>
<tr>
<td><strong>HHMM: par+sib, sect 23</strong></td>
<td><strong>79.7</strong></td>
<td><strong>80.4</strong></td>
<td><strong>80.1</strong></td>
<td><strong>0.6</strong></td>
</tr>
<tr>
<td>Henderson’04, non-det., sect 0</td>
<td></td>
<td></td>
<td>89.8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LP</th>
<th>LR</th>
<th>F</th>
<th>fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>no punctuation: (≤ 120 wds)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R’01: par+sib, sect 23–24</td>
<td>77.4</td>
<td>75.2</td>
<td>–</td>
<td>0.1</td>
</tr>
<tr>
<td><strong>HHMM: par+sib, sect 23–24</strong></td>
<td><strong>77.6</strong></td>
<td><strong>76.8</strong></td>
<td><strong>77.2</strong></td>
<td><strong>0.4</strong></td>
</tr>
</tbody>
</table>
Quintuple center-embedding

Here’s one of the 16 depth-five sentences in the corpus:

```
S
  on ... prosperity
  S
    SBAR
      if
      S
        S
          America can
          VP
            keep up
              VP
                NP
                  the present situation
              NP
                  her markets open
            VP
              for another 15 years ...
    S
      and Japan can grow and ...
  S
    then ...

Left-/right-corner won’t undo zig-zags. Need them to untangle referents.
```
Conclusion

Right-corner transform explains parsing w/in human-like memory limits.
Bounded memory HHMM model mostly safe, in terms of coverage.
But, no big windfall in accuracy.

Future work:

▶ Lexicalization / vector-space semantics
▶ Smarter strategy for deferring composition if memory not used up
▶ Smoothing, backoff
▶ Estimate joint probabilities over entire columns