Mildly Context-Sensitive Grammar Formalisms:

LCFRS Parsing

Laura Kallmeyer

Sommersemester 2011

Overview

1. Ranges
2. CYK Parsing
3. Incremental Earley Parsing
   (a) Deduction Rules
   (b) Filters

Ranges (1)

- During parsing we have to link the terminals and variables in our LCFRS rules to portions of the input string.
- These can be characterized by their start and end positions.
- A *range* is an pair of indices that characterizes the span of a component within the input and a range vector characterizes a tuple in the yield of a non-terminal.
- The range instantiation of a rule specifies the computation of an element from the lefthand side yield from elements of in the yields of the right-hand side non-terminals based on the corresponding range vectors.

Ranges (2)

Example: Rule $A(aXa, bYb) \rightarrow B(X)C(Y)$ and input string $ababab cb$.

We assume without loss of generality that our LCFRSs are monotone and ε-free. Furthermore, because of the linearity, the components of a tuple in the yield of an LCFRS non-terminal are necessarily non-overlapping. Then, given this input, we have the following possible instantiations for this rule:

$$A({}_0aba_3, {}_3bab_6) \rightarrow B({}_1b_2, {}_4a_5) \quad A({}_0aba_3, {}_3babcb_8) \rightarrow B({}_1b_2, {}_4abc_7)$$
$$A({}_0aba_3, {}_5bcb_8) \rightarrow B({}_1b_2, {}_6c_7) \quad A({}_0ababa_5, {}_5bcb_8) \rightarrow B({}_1bab_4, {}_6c_7)$$
$$A({}_2aba_5, {}_5bcb_8) \rightarrow B({}_3b_4, {}_6c_7)$$

Ranges (3)

Definition 1 (Range) *Let $w \in T^*$ be a word with $w = w_1 \dots w_n$ where $w_i \in T$ for $1 \leq i \leq n$.*

1. $Pos(w) := \{0, \dots, n\}$.
2. *We call a pair $\langle l, r\rangle \in Pos(w) \times Pos(w)$ with $l \leq r$ a* range *in w. Its* yield *$\langle l, r\rangle(w)$ is the substring $w_{l+1} \dots w_r$.*
3. *For two ranges $\rho_1 = \langle l_1, r_1\rangle, \rho_2 = \langle l_2, r_2\rangle$, if $r_1 = l_2$, then the concatenation of ρ_1 and ρ_2 is $\rho_1 \cdot \rho_2 = \langle l_1, r_2\rangle$; otherwise $\rho_1 \cdot \rho_2$ is undefined.*
4. *Two ranges $\langle l_1, r_1\rangle, \langle l_2, r_2\rangle$ are* overlapping *if*
   (a) *either $l_1 \leq l_2 < r_1$ and $l_1 < r_2$*
   (b) *or $l_1 < r_2 \leq r_1$ and $l_2 < r_1$.*

Ranges (3)

Definition 2 (Range vector)

Let $w \in T^*$.

1. *A $\vec{\rho} \in (Pos(w) \times Pos(w))^k$ is a k-dimensional* range vector *for w iff $\vec{\rho} = \langle\langle l_1, r_1\rangle, \dots, \langle l_k, r_k\rangle\rangle$ where $\langle l_i, r_i\rangle$ is a range in w for $1 \leq i \leq k$.*
2. *For a k-dimensional range vector $\vec{\rho}$ for w we define the denotation of $\vec{\rho}$ as $\vec{\rho}(w) := \langle\langle l_1, r_1\rangle(w), \dots, \langle l_k, r_k\rangle(w)\rangle$.*

A range vector $\vec{\rho}$ is called simple *iff its elements are pairwise non-overlapping.*

Ranges (3)

Definition 3 (Range instantiation, [Boullier, 2000]) *Let $G = (N, T, V, P, S)$ be a LCFRS. For a given rule $\gamma = A(\vec{\alpha}) \to A_1(\vec{\alpha_1}) \cdots A_m(\vec{\alpha_m}) \in P$ $(0 \leq m)$,*

1. *a* range instantiation *with respect to a string $w = t_1 \dots t_n$ is a function $f : \{t' \mid t'$ is an occurrence of some $t \in T$ in the clause$\} \cup V \cup \{Eps_i \mid 1 \leq i \leq dim(A), \vec{\alpha}(i) = \varepsilon\} \to \{\langle i, j\rangle \mid i \leq j, i, j \in \mathbb{N}\}$ such that*
   a) *for all occurrences t' of a $t \in T$ in $\vec{\alpha}$, $f(t')(w) = t$,*
   b) *for all $X \in V$, $f(X) = \langle j, k\rangle$ for some $0 \leq j \leq k \leq n$,*
   c) *for all x, y adjacent in one of the elements of $\vec{\alpha}$ there are i, j, k with $f(x) = \langle i, j\rangle, f(y) = \langle j, k\rangle$; we define then $f(xy) = \langle i, k\rangle$,*

   d) *for all $Eps \in \{Eps_i \mid 1 \leq i \leq dim(A), \vec{\alpha}(i) = \varepsilon\}$, there is a j, $0 \leq j \leq n$ with $f(Eps) = \langle j, j\rangle$; we define then for every ε-argument $\vec{\alpha}(i)$ that $f(\vec{\alpha}(i)) = f(Eps_i)$;*

2. *if f is an instantiation of a γ, then $A(f(\vec{\alpha})) \to A_1(f(\vec{\alpha_1})) \cdots A_m(f(\vec{\alpha_m}))$ is an* instantiated rule *where $f(\langle x_1, \dots, x_k\rangle) = \langle f(x_1), \dots, f(x_k)\rangle$.*

CYK Parsing (1)

First introduced in [Seki et al., 1991]; deduction-based definition in, e.g., [Kallmeyer and Maier, 2010].

Idea: Once all predicates in the RHS of a rules have been found, complete LHS, more precisely:

- We start with the terminal symbols: whenever we can find a range instantiation of a rule with rhs ε, we conclude that this rule can be applied (*scan* operation),
- We traverse the derivation tree bottom-up: whenever, for a range instantiation of a rule, all pairs of non-terminal symbol and range vector in the rhs have been found, we conclude that this rule can be applied and the lhs of the instantiated rule is deduced (*complete* operation).
- Our input w is in the language iff S with range vector $\langle\langle 0, n\rangle\rangle$ is in the final set of results that we have deduced.

CYK Parsing (2)

Deduction rules:

Items $[A, \vec{\rho}]$ with $A \in N$, $\vec{\rho}$ is a $dim(A)$-dimensional range vector in w.

Axioms: $\dfrac{}{[A, \vec{\rho}]}$ $A(\vec{\rho}) \rightarrow \varepsilon$ a range instantiated rule

Complete: $\dfrac{[A_1, \vec{\rho_1}], \dots, [A_m, \vec{\rho_m}]}{[A, \vec{\rho}]}$ $A(\vec{\rho}) \rightarrow A_1(\vec{\rho_1}), \dots, A_m(\vec{\rho_m})$ a range instantiated rule

Goal item: $[S, \langle\langle 0, n\rangle\rangle]$

CYK Parsing (3)

Deduction rules for binarized ε-free grammars where, without loss of generality, either the lhs contains a single terminal and the rhs is ε or the rule contains only variables:

Items and goal as before.

Scan: $\dfrac{}{[A, \langle\langle i, i+1\rangle\rangle]}$ $A(w_{i+1}) \rightarrow \varepsilon \in P$

Unary: $\dfrac{[B, \vec{\rho}]}{[A, \vec{\rho}]}$ $A(\vec{\alpha}) \rightarrow B(\vec{\alpha}) \in P$

Binary: $\dfrac{[B, \vec{\rho_B}], [C, \vec{\rho_C}]}{[A, \vec{\rho_A}]}$ $A(\vec{\rho_A}) \rightarrow B(\vec{\rho_B})C(\vec{\rho_C})$ is a range instantiated rule

CYK Parsing (4)

Complexity of CYK parsing with binarized LCFRSs:

We have to consider the maximal number of possible applications of the complete rule.

Binary: $\dfrac{[B, \vec{\rho_B}], [C, \vec{\rho_C}]}{[A, \vec{\rho_A}]}$ $A(\vec{\rho_A}) \rightarrow B(\vec{\rho_B})C(\vec{\rho_C})$ is a range instantiated rule

If k is the maximal fan-out in the LCFRS, we have maximal $2k$ range boundaries in each of the antecedent items of this rule. For variables X_1, X_2 being in the same lhs side argument of the rule, X_1 left of X_2 and no other variables in between, the right boundary of X_1 is the left boundary of X_2. In the worst case, A, B, C all have fan-out k and each lhs argument contains two variables. This gives $3k$ independent range boundaries and consequently a time complexity of $\mathcal{O}(n^{3k})$ for the entire algorithm.

Incremental Earley Parsing

Strategy:

- Process LHS arguments incrementally, starting from an S-rule
- Whenever we reach a variable, move into rule of correponding rhs non-terminal (**predict** or **resume**).
- Whenever we reach the end of an argument, **suspend** the rule and move into calling parent rule.
- Whenever we reach the end of the last argument **convert** item into a passive one and **complete** parent item.

This parser is described in [Kallmeyer and Maier, 2009] and inspired by the Thread Automata in [Villemonte de La Clergerie, 2002]

Incremental Earley Parsing: Items

- **Passive items**: $[A, \vec{\rho}]$ where A is a non-terminal of fan-out k and $\vec{\rho}$ is a range vector of fan-out k
- **Active items**:

$$[A(\vec{\phi}) \rightarrow A_1(\vec{\phi_1}) \dots A_m(\vec{\phi_m}), pos, \langle i, j\rangle, \vec{\rho}]$$

where

- $A(\vec{\phi}) \rightarrow A_1(\vec{\phi_1}) \dots A_m(\vec{\phi_m}) \in P$;
- $pos \in \{0, \dots, n\}$: We have reached input position pos;
- $\langle i, j\rangle \in \mathbb{N}^2$: We have reached the jth element of ith argument (dot position);
- $\vec{\rho}$ is a range vector containing variable and terminal bindings. All elements are initialized to "?", an initialized vector is called $\vec{\rho}_{init}$.

Incremental Earley Parsing: Example (1)

$S(X_1X_2) \longrightarrow A(X_1, X_2)$ $\quad A(aX_1, bX_2) \longrightarrow A(X_1, X_2)$ $\quad A(a, b) \longrightarrow \varepsilon$

Parsing trace for input $w = aabb$:

	pos	item	$\vec{\rho}$	
1	0	$S(\bullet X_1X_2) \longrightarrow A(X_1, X_2)$	$(?, ?)$	axiom
2	0	$A(\bullet aX_1, bX_2) \longrightarrow A(X_1, X_2)$	$(?, ?, ?, ?)$	predict, 1
3	0	$A(\bullet a, b) \longrightarrow \varepsilon$	$(?, ?)$	predict, 1
4	1	$A(a \bullet X_1, bX_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 1\rangle, ?, ?, ?)$	scan, 2
5	1	$A(a\bullet, b) \longrightarrow \varepsilon$	$(\langle 0, 1\rangle, ?)$	scan, 3
6	1	$A(\bullet aX_1, bX_2) \longrightarrow A(X_1, X_2)$	$(?, ?, ?, ?)$	predict, 4
7	1	$A(\bullet a, b) \longrightarrow \varepsilon$	$(?, ?)$	predict 4
8	1	$S(X_1 \bullet X_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 1\rangle, ?)$	susp. 5, 1
9	1	$A(a, \bullet b) \longrightarrow \varepsilon$	$(\langle 0, 1\rangle, ?)$	resume 5, 8

Incremental Earley Parsing: Example (2)

10	2	$A(a \bullet X_1, bX_2) \longrightarrow A(X_1, X_2)$	$(\langle 1, 2\rangle, ?, ?, ?)$	scan 6
11	2	$A(a\bullet, b) \longrightarrow \varepsilon$	$(\langle 1, 2\rangle, ?)$	scan 7
12	2	$A(\bullet aX_1, bX_2) \longrightarrow A(X_1, X_2)$	$(?, ?, ?, ?)$	predict 10
13	2	$A(\bullet a, b) \longrightarrow \varepsilon$	$(?, ?)$	predict 10
14	2	$A(aX_1\bullet, bX_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 1\rangle, \langle 1, 2\rangle, ?, ?)$	susp. 11, 4
15	2	$S(X_1 \bullet X_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 2\rangle, ?)$	susp. 14, 1
16	2	$A(aX_1, \bullet bX_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 1\rangle, \langle 1, 2\rangle, ?, ?)$	resume 14, 15
17	3	$A(aX_1, b \bullet X_2) \longrightarrow A(X_1, X_2)$	$(\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, ?)$	scan 16
18	3	$A(a, \bullet b) \longrightarrow \varepsilon$	$(\langle 1, 2\rangle, ?)$	resume 11, 17

Incremental Earley Parsing: Example (3)

19	4	$A(a, b\bullet) \longrightarrow \varepsilon$	$(\langle 1,2\rangle, \langle 3,4\rangle)$	scan 18
20	4	$A(\langle 1,2\rangle, \langle 3,4\rangle)$		convert 19
21	4	$A(aX_1, bX_2\bullet) \longrightarrow A(X_1, X_2)$	$(\langle 0,1\rangle, \langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle)$	compl. 17, 20
22	4	$A(\langle 0,2\rangle, \langle 2,4\rangle)$		convert 21
23	4	$S(X_1X_2\bullet) \longrightarrow A(X_1, X_2)$	$(\langle 0,2\rangle, \langle 2,4\rangle)$	compl. 15, 22
24	4	$S(\langle 0,4\rangle)$		convert 23

Incremental Earley Parsing: Deduction Rules

- Notation:
  - $\vec{\rho}(X)$: range bound to variable X.
  - $\vec{\rho}(\langle i, j\rangle)$: range bound to jth element of ith argument on LHS.
- Applying a range vector $\vec{\rho}$ containing variable bindings for given rule c to the argument vector of the lefthand side of c means mapping the ith element in the arguments to $\vec{\rho}(i)$ and concatenating adjacent ranges. The result is defined iff every argument is thereby mapped to a range.

Incremental Earley Parsing: Initialize, Goal item

Initialize: $\dfrac{}{[S(\vec{\phi}) \to \vec{\Phi}, 0, \langle 1, 0\rangle, \vec{\rho}_{init}]}\quad S(\vec{\phi}) \to \vec{\Phi} \in P$

Goal Item: $[S(\vec{\phi}) \to \vec{\Phi}, n, \langle 1, j\rangle, \psi]$ with $|\vec{\phi}(1)| = j$ (i.e., dot at the end of lhs argument).

Incremental Earley Parsing: Scan

If next symbol after dot is next terminal in input, scan it.

Scan: $\dfrac{[A(\vec{\phi}) \to \vec{\Phi}, pos, \langle i, j\rangle, \vec{\rho}]}{[A(\vec{\phi}) \to \vec{\Phi}, pos+1, \langle i, j+1\rangle, \vec{\rho}']}\quad \vec{\phi}(i, j+1) = w_{pos+1}$

where $\vec{\rho}'$ is $\vec{\rho}$ updated with $\vec{\rho}(\langle i, j+1\rangle) = \langle pos, pos+1\rangle$.

Incremental Earley Parsing: Predict

Whenever our dot is left of a variable that is the first argument of some rhs non-terminal B, we predict new B-rules:

Predict: $$\frac{[A(\vec{\phi}) \to \dots B(X, \dots) \dots, pos, \langle i, j\rangle, \vec{\rho}_A]}{[B(\vec{\psi}) \to \vec{\Psi}, pos, \langle 1, 0\rangle, \vec{\rho}_{init}]}$$

where $\vec{\phi}(i, j+1) = X, B(\vec{\psi}) \to \vec{\Psi} \in P$

Incremental Earley Parsing: Suspend

Suspend:
$$\frac{[B(\vec{\psi}) \to \vec{\Psi}, pos', \langle i, j\rangle, \vec{\rho}_B], [A(\vec{\phi}) \to \dots B(\vec{\xi}) \dots, pos, \langle k, l\rangle, \vec{\rho}_A]}{[A(\vec{\phi}) \to \dots B(\vec{\xi}) \dots, pos', \langle k, l+1\rangle, \vec{\rho}]}$$

where

- the dot in the antecedent A-item precedes the variable $\vec{\xi}(i)$,
- $|\vec{\psi}(i)| = j$ (ith argument has length j, i.e., is completely processed),
- $|\vec{\psi}| < i$ (ith argument is not the last argument of B),
- $\vec{\rho}_B(\vec{\psi}(i)) = \langle pos, pos'\rangle$
- and for all $1 \leq m < i$: $\vec{\rho}_B(\vec{\psi}(m)) = \vec{\rho}_A(\vec{\xi}(m))$.

$\vec{\rho}$ is $\vec{\rho}_A$ updated with $\vec{\rho}_A(\vec{\xi}(i)) = \langle pos, pos'\rangle$.

Incremental Earley Parsing: Convert

Whenever we arrive at the end of the last argument, we convert the item into a passive one:

Convert:
$$\frac{[B(\vec{\psi}) \to \vec{\Psi}, pos, \langle i, j\rangle, \vec{\rho}_B]}{[B, \rho]} \quad |\vec{\psi}(i)| = j, |\vec{\psi}| = i, \vec{\rho}_B(\vec{\psi}) = \rho$$

Incremental Earley Parsing: Complete

Whenever we have a passive B item we can use it to move the dot over the variable of the last argument of B in a parent A-rule:

Complete: $$\frac{[B, \vec{\rho}_B], [A(\vec{\phi}) \to \dots B(\vec{\xi}) \dots, pos, \langle k, l\rangle, \vec{\rho}_A]}{[A(\vec{\phi}) \to \dots B(\vec{\xi}) \dots, pos', \langle k, l+1\rangle, \vec{\rho}]}$$ where

- the dot in the antecedent A-item precedes the variable $\vec{\xi}(|\vec{\rho}_B|)$,
- the last range in $\vec{\rho}_B$ is $\langle pos, pos'\rangle$,
- and for all $1 \leq m < |\vec{\rho}_B|$: $\vec{\rho}_B(m) = \vec{\rho}_A(\vec{\xi}(m))$.

$\vec{\rho}$ is $\vec{\rho}_A$ updated with $\vec{\rho}_A(\vec{\xi}(|\vec{\rho}_B|)) = \langle pos, pos'\rangle$.

Incremental Earley Parsing: Resume

Whenever we are left of a variable that is not the first argument of one of the rhs non-terminals, we resume the rule of the rhs non-terminal.

$$\textbf{Resume}: \frac{[A(\vec{\phi}) \to \ldots B(\vec{\xi}) \ldots, pos, \langle i, j\rangle, \vec{\rho}_A], \quad [B(\vec{\psi}) \to \vec{\Psi}, pos', \langle k-1, l\rangle, \vec{\rho}_B]}{[B(\vec{\psi}) \to \vec{\Psi}, pos, \langle k, 0\rangle, \vec{\rho}_B]}$$

where

- $\vec{\phi}(i, j+1) = \vec{\xi}(k), k > 1$ (the next element is a variable that is the kth element in $\vec{\xi}$, i.e., the kth argument of B),
- $|\vec{\psi}(k-1)| = l$, and
- $\vec{\rho}_A(\vec{\xi}(m)) = \vec{\rho}_B(\vec{\psi}(m))$ for all $1 \leq m \leq k-1$.

Incremental Earley Parsing: Filters

- Filters can be applied to decrease the number of items in the chart
- A filter is an additional condition on the form of items.
- E.g., in a ε-free grammar, the number of variables in the part of the lefthand side arguments of a rule that has not been processed yet must be lower or equal to the length of the remaining input.

Incremental Earley Parsing: Remaining Input Length Filter

- In ε-free grammars each variable must cover at least one input symbol.
- i input symbols left implies no prediction of a clause with more than i variables or terminals on LHS since no instantiation is possible
- Condition on active items, can be applied with predict, resume, suspend and complete

An item $[A(\vec{\phi}) \to A_1(\vec{\phi_1}) \ldots A_m(\vec{\phi_m}), pos, \langle i, j\rangle, \vec{\rho}]$ satisfies the **length filter** iff

$$(n - pos) \geq (|\vec{\phi}(i)| - j) + \Sigma_{k=i+1}^{dim(A)} |\vec{\phi}(k)|$$

Incremental Earley Parsing: Preterminal Filter (1)

- Check for the presence of (pre)terminals in the predicted part of a clause in the remaining input, and
- check that terminals appear in the predicted order and that distance between two of them is at least the number of variables/terminals in between.

continued…

Incremental Earley Parsing: Preterminal Filter (2)

In other words, an active item $[A(\vec{\phi}) \rightarrow A_1(\vec{\phi_1}) \dots A_m(\vec{\phi_m}), pos, \langle i, j \rangle, \vec{\rho}]$ satisfies the **preterminal filter** iff we can find an injective mapping $f_T : \text{Term} = \{\langle k, l \rangle \mid \vec{\phi}(k, l) \in T$ and either $k > i$ or $(k = i$ and $l > j)\} \rightarrow \{pos + 1, \dots, n\}$ such that

1. $w_{f_T(\langle k,l \rangle)} = \vec{\phi}(k, l)$ for all $\langle k, l \rangle \in Term$;
2. for all $\langle k_1, l_1 \rangle, \langle k_2, l_2 \rangle \in Term$ with $k_1 = k_2$ and $l_1 < l_2$: $f_T(\langle k_2, l_2 \rangle) \geq f_T(\langle k_1, l_1 \rangle) + (l_2 - l_1)$;
3. for all $\langle k_1, l_1 \rangle, \langle k_2, l_2 \rangle \in Term$ with $k_1 < k_2$: $f_T(\langle k_2, l_2 \rangle) \geq f_T(\langle k_1, l_1 \rangle) + (|\vec{\phi}(k_1)| - l_1) + \Sigma_{k=k_1+1}^{k_2-1} |\vec{\phi}(k)| + l_2$.

References

[Boullier, 2000] Boullier, P. (2000). Range Concatenation Grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT2000)*, pages 53–64, Trento, Italy.

[Kallmeyer and Maier, 2009] Kallmeyer, L. and Maier, W. (2009). An incremental Earley parser for simple Range Concatenation Grammar. In *Proceedings of IWPT 2009*.

[Kallmeyer and Maier, 2010] Kallmeyer, L. and Maier, W. (2010). Data-driven parsing with probabilistic Linear Context-Free Rewriting Systems. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China.

[Seki et al., 1991] Seki, H., Matsumura, T., Fujii, M., and Kasami, T. (1991). On multiple context-free grammars. *Theoretical

Computer Science, 88(2):191–229.

[Villemonte de La Clergerie, 2002] Villemonte de La Clergerie, E. (2002). Parsing mildly context-sensitive languages with thread automata. In *Proc. of COLING'02*.